



TUGAS AKHIR - TE 141599

**Perencanaan Lintasan Mobile Robot Menggunakan
Metode RBF (Radial Basis Function) dan Telusur
Trayektori dengan Kontroler PID-Fuzzy**

Michael Calvin Rendi Arianto
NRP. 2215 105 048

Dosen Pembimbing
Ir. Rusdhianto Effendi A.K., M.T

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 141599

Path Planning Mobile Robot Using RBF (Radial Basis Function) Method and Trajectory tracking with PID-Fuzzy Controller

Michael Calvin Rendi Arianto
NRP. 2215 105 048

Supervisor
Ir. Rusdhianto Effendi A.K., M.T

***DEPARTEMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Industrial
Sepuluh Nopember Insitute of Technology
Surabaya 2017***

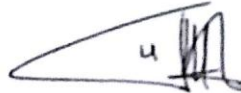
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul **“Perencanaan Lintasan Mobile Robot Menggunakan Metode RBF (*Radial Basis Function*) dan Telusur Trayektori dengan Kontroler *PID-Fuzzy*”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017

A handwritten signature in black ink, appearing to be 'Michael Calvin', written over a horizontal line.

Michael Calvin
NRP. 2215 105 048

(halaman ini sengaja dikosongkan)

**PERENCANAAN LINTASAN MOBILE ROBOT
MENGUNAKAN METODE RBF (RADIAL BASIS FUNCTION)
DAN TELUSUR TRAYEKTORI DENGAN KONTROLER PID-
FUZZY**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro
Pada
Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,


Ir. Rusdhianto Effendi A.K., M.T.
NIP. 1952 04 24 1985 03 1001



(halaman ini sengaja dikosongkan)

**PERENCANAAN LINTASAN *MOBILE ROBOT*
MENGUNAKAN METODE RBF (*RADIAL BASIS FUNCTION*)
DAN TELUSUR TRAYEKTORI DENGAN KONTROLER *PID-FUZZY***

Nama : Michael Calvin Rendi Arianto
NRP : 2215 105 048

Pembimbing I : Ir. Rusdhianto Effendi A.K., M.T.
NIP : 195204241985021001

ABSTRAK

Pada perkembangan teknologi terutama dalam bidang robotik, peran *mobile robot* akan sangat berguna bagi kehidupan manusia. Seiring fungsi *mobile robot* yang dapat membantu pekerjaan manusia yang berbahaya, kotor, berulang ulang atau yang sulit untuk dilakukan. Salah satunya adalah perancangan lintasan *mobile robot* agar dapat mencapai titik atau target. Pada tugas akhir ini *mobile robot* yang digunakan adalah Quanser Qbot. Metode yang digunakan untuk perancangan lintasan adalah *radial basis function*. Dimana kita dapat memperkirakan lintasan yang akan dilalui *mobile robot*. Serta kontroler *PID fuzzy* agar *mobile robot* dapat menghindari halangan dengan lebih lancar (*smooth*).

Hasil pengujian pada simulasi dan implementasi *Radial Basis Function* mampu memperoleh lintasan tanpa menabrak *obstacles* yaitu pada simulasi dengan 2 *obstacles* dan dengan 3 *obstacles*. Dengan titik awal (x,y) yaitu (0,0) hingga mencapai titik akhir (x,y) yaitu (200,0). Jarak yang ditempuh sejauh 2 meter. Dengan hasil implementasi lintasan dengan 2 *obstacles* dilalui dengan *set point* sebesar 100, waktu yang dibutuhkan 68,70 detik, sedangkan hasil implementasi lintasan dengan 3 *obstacles* membutuhkan waktu 59,4 detik mulai dari posisi awal hingga menuju titik target.

Kata kunci : *Mobile Robot, Smooth, Radial Basis Function, PID fuzzy, obstacles*

(halaman ini sengaja dikosongkan)

**PATH PLANNING MOBILE ROBOT USING RBF (RADIAL BASIS
FUNCTION) METHOD AND TRAJECTORY TRACKING WITH
PID-FUZZY CONTROLLER**

Name : Michael Calvin Rendi Arianto
NRP : 2215 105 048

Supervisor I : Ir. Rusdhianto Effendi A.K., M.T.
ID Number : 195204241985021001

ABSTRACT

In technological developments especially in the field of robotics, the role of mobile robot will be very useful for human life. Along with mobile robot function that can help human work that is dangerous, dirty, repetitive or difficult to do. One of them is the design of mobile robot path in order to reach the point or target. In this final task mobile robot that used is Quanser Qbot. The method used for path design is radial basis function. Where we can estimate the path that will be passed by mobile robot. And fuzzy PID controller for mobile robot can avoid obstruction more smoothly (smooth).

The test results on the simulation and implementation of Radial Basis Function are able to obtain path without obstacles ie simulation pattern 1 and 2. With the starting point (x, y) is (0, 0) to reach the end point (x, y) at (200, 0). Distance traveled as far as 2 meters. With the result of path with two obstacles and set point 100, is traversed with time required 68,70 seconds, while the result of path with three obstacles takes 59,4 seconds starting from the initial position to the target point.

Keywords : Mobile Robot, Smooth, Radial Basis Function, PID fuzzy, obstacles

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT karena berkat rahmat dan hidayah-Nya penulis dapat menyusun dan menyelesaikan laporan Tugas Akhir dengan judul:

Perencanaan Lintasan *Mobile Robot* Menggunakan *Metode RBF (Radial Basis Function)* dan Telusur Trayektori dengan Kontroler *PID-Fuzzy*

Penulis ingin berterima kasih kepada beberapa pihak yang telah membantu dalam penyusunan dan penyelesaian dari Tugas Akhir ini, diantaranya:

1. Kedua orang tua, Soni Rumanto dan Rita Susanti yang selalu memberikan dukungan, semangat, materi dan doa kepada penulis.
2. Bapak Ir. Rusdhianto Effendie .A.K, MT atas bantuan dan bimbingan hingga Tugas Akhir ini terselesaikan.
3. Para dosen penguji Tugas Akhir yang telah memberi masukan dan saran kepada penulis dalam penyelesaian Tugas Akhir.
4. Semua teman yang ada di Teknik Sistem Pengaturan khususnya anggota kelompok Tugas Akhir Tegar Wangi dan Muhammad Fajar atas bantuan dan kerjasama yang telah diberikan.
5. Rekan – rekan mahasiswa Teknik Lektro serta rekan kos Gebang Wetan blok 25 C yang telah menampung penulis selama menjadi mahasiswa.

Harapan besar penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Jurusan Teknik Elektro pada khususnya. Penulis juga mengharapkan kritik dan saran atas buku Tugas Akhir ini karena penulis menyadari bahwa dalam penyusunan Tugas Akhir ini banyak terdapat kekurangan.

Surabaya, Juni 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Metodologi	2
1.5 Sistematika	3
1.6 Relevansi	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Tinjauan Pustaka	5
2.2 Mobile Robot Qbot	6
2.2.1 Baterai Mobile Robot Qbot	7
2.2.2 Sensor Inframerah dan Sonar	8
2.3 QUARC	9
2.4 Radial Basis Function	12
2.5 Kontroler <i>PID Fuzzy</i>	13
2.5.1 Logika <i>Fuzzy</i>	13
2.5.2 PID (Proportional Integrative Derivative)	20
BAB 3 PERANCANGAN SISTEM	23
3.1 Diagram Blok Sistem	23
3.2 Model Matematika <i>Mobile Robot Qbot</i>	24
3.2.1 Differential Drive Kinematics	24
3.2.2 Lokasi ICC	26
3.2.3 Forward Kinematik	26
3.2.4 Inverse Kinematics	28
3.3 Perancangan <i>Path Planning</i> dengan Metode <i>Radial Basis Function</i>	29
3.4 Perancangan Kontroler <i>PID-Fuzzy</i>	30
3.5 Perancangan <i>Software</i>	32
3.5.1 Perancangan <i>Software Simulasi</i>	32

3.5.2	Perancangan <i>Software</i> Implementasi	34
BAB 4	HASIL DAN ANALISA	37
4.1	Simulasi Perencanaan Lintasan dengan <i>Radial Basis Function</i> 37	
4.1.1	Simulasi Perencanaan lintasan RBF dengan 2 <i>obstacles</i> 37	
4.1.2	Simulasi Perencanaan lintasan RBF dengan 3 <i>obstacles</i> 39	
4.2	Implementasi Trayektori dengan Kontroler PID Fuzzy	41
4.2.1	Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 <i>Obstacles</i>	41
4.2.2	Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 <i>Obstacles</i>	45
4.3	Implementasi Kontroler <i>PID Fuzzy</i>	48
BAB 5	PENUTUP	51
5.1	Kesimpulan	51
5.2	Saran	51
	DAFTAR PUSTAKA	53
	LAMPIRAN	55
	RIWAYAT PENULIS	69

TABLE OF CONTENTS

TITLE PAGE	iii
STATEMENT OF FINAL PROJECT	v
APPROVALSHEET	vii
ABSTRACT	ix
ABSTRACT	xi
PREFACE	xiii
DAFTAR ISI	xv
LIST OF FIGURE	xix
LIST OF TEBLE	xxi
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Problems	2
1.3 Intention	2
1.4 Methodology	2
1.5 Sistematics	3
1.6 Research objective	3
CHAPTER 2 BASIC THEORY	5
2.1 Basic Theory	5
2.2 Mobile Robot Qbot	6
2.2.1 Battery of Mobile Robot Qbot.....	7
2.2.2 Sensor Infrared and Sonar.....	8
2.3 QUARC	9
2.4 Radial Basis Function	12
2.5 Controller <i>PID Fuzzy</i>	13
2.5.1 <i>Fuzzy Logic</i>	13
2.5.2 PID (Proportional Integrative Derivative)	20
CHAPTER 3 SYSTEM DESIGN	23
3.1 Diagram Block System	23
3.2 Model Mathematics <i>Mobile Robot Qbot</i>	24
3.2.1 Differential Drive Kinematics	24
3.2.2 Location ICC	26
3.2.3 Forward Kinematik.....	26
3.2.4 Inverse Kinematics	28
3.3 <i>Path Planning</i> with Metode <i>Radial Basis Function</i>	29
3.4 Controller Planning <i>PID-Fuzzy</i>	30
3.5 Planning <i>Software</i>	32
3.5.1 Planning <i>Software Simulation</i>	32
3.5.2 Planning <i>Software Implementation</i>	34

CHAPTER 4 RESULT AND ANALYSIS	37
4.1 Planning Simulation Lintasan with <i>Radial Basis Function</i>	37
4.1.1 Simulation path planning RBF with 2 <i>obstacles</i>	37
4.1.2 Simulation path planning RBF with 3 obstacles	39
4.2 Implementation Trajectory with PID Fuzzy Controller	41
4.2.1 Implementation Trajectory with PID Fuzzy Controller with 2 Obstacles	41
4.2.2 Implementation Trajectory with PID Fuzzy Controller with 3 Obstacles	45
4.3 Implementation <i>PID Fuzzy</i> Controller	48
CHAPTER 5 CLOSING	51
5.1 Conclusion.....	51
5.2 Advice.....	51
BIBLIOGRAPHY	53
ENCLOSURE	55
BIOGRAPHY	69

DAFTAR GAMBAR

Gambar 2.1 <i>Mobile Robot</i> Qbot [3].....	6
Gambar 2.2 Hierarki Komunikasi <i>Mobile Robot</i> Qbot [3]	7
Gambar 2.3 Baterai <i>Mobile Robot</i> Qbot [3]	7
Gambar 2.4 Letak Baterai pada <i>Mobile Robot</i> Qbot [3]	8
Gambar 2.5 Sensor Inframerah SHARP 2Y0A02 [3].....	8
Gambar 2.6 Sensor Sonar MaxSonar-EZ0 [3].....	9
Gambar 2.7 Struktur RBF Network.....	12
Gambar 2.8 Bentuk-bentuk <i>Membership Function</i>	15
Gambar 2.9 Struktur Kontroler <i>Fuzzy</i> [4].....	17
Gambar 2.10 Struktur Fungsi Keanggotaan <i>Fuzzy</i>	18
Gambar 3.1 Diagram Blok Fungsional	23
Gambar 3.2 Kinematik dari Differential Robot [6]	25
Gambar 3.3 Forward Kinematics Relative to ICC.....	27
Gambar 3.4 Diagram Blok Kontroler <i>PID – Fuzzy</i>	30
Gambar 3.5 Himpunan Keanggotaan masukan <i>error</i>	31
Gambar 3.6 Blok <i>Simulink</i> Simulasi <i>Mobile Robot</i> Qbot.....	33
Gambar 3.7 Simulasi Perencanaan Lintasan <i>Radial Basis Function</i> ...	34
Gambar 3.8 Implementasi Perencanaan Lintasan <i>Radial Basis Function</i>	35
Gambar 3.9 Blok <i>Simulink</i> Implementasi Qbot.....	36
Gambar 4.1 Simulasi Perencanaan lintasan RBF dengan 2 <i>obstacles</i> .	37
Gambar 4.2 Posisi Robot Training Lintasan 1.....	38
Gambar 4.3 Hasil <i>Trajectory</i> Perencanaan lintasan RBF dengan 2 <i>obstacles</i>	38
Gambar 4.4 Simulasi Perencanaan lintasan RBF dengan 3 <i>obstacles</i> .	39
Gambar 4.5 Posisi Robot <i>Training</i> Lintasan 2	40
Gambar 4.6 Hasil <i>Trajectory</i> Perencanaan lintasan RBF dengan 3 <i>obstacles</i>	40
Gambar 4.7 Hasil <i>Track</i> Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 <i>Obstacles</i>	41
Gambar 4.8 Kecepatan Roda Kanan Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 <i>Obstacles</i>	42
Gambar 4.9 Kecepatan Roda Kiri Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 <i>Obstacles</i>	42

Gambar 4.10 Respons ω_x Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles	43
Gambar 4.11 Respons ω_y Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles	43
Gambar 4.12 Hasil <i>Track</i> Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles	45
Gambar 4.13 Kecepatan Roda Kanan Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles	45
Gambar 4.14 Kecepatan Roda Kiri Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles	46
Gambar 4.15 Respons ω_x Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles	46
Gambar 4.16 Respons ω_y Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles	47

DAFTAR TABEL

Tabel 2.1 Spesifikasi Model <i>Mobile Robot Qbot</i>	6
Tabel 2.2 Deskripsi Blok Set untuk <i>Mobile Robot Qbot</i>	9
Tabel 2.3 Deskripsi Blok QUARC pada <i>Mobile Robot Qbot</i>	10
Tabel 3.1 Rule Base dengan Delta Error Lambat	31
Tabel 3.2 Rule Base dengan Delta Error Sedang	31
Tabel 3.3 Rule Base dengan Delta Error Cepat	32
Tabel 4.1 Kecepatan Roda Kanan	44
Tabel 4.2 Kecepatan Roda Kiri	44
Tabel 4.3 Kecepatan pada sumbu x	44
Tabel 4.4 Kecepatan pada sumbu y	44
Tabel 4.5 Kecepatan Roda Kanan	47
Tabel 4.6 Kecepatan Roda Kiri	48
Tabel 4.7 Kecepatan pada sumbu x	48
Tabel 4.8 Kecepatan pada sumbu y	48
Tabel 4.9 Hasil Implementasi <i>PID Fuzzy</i> Lintasan 1	49
Tabel 4.10 Hasil Implementasi <i>PID Fuzzy</i> Lintasan 2	49

(halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

Pada bab satu akan dibahas mengenai latar belakang, permasalahan, tujuan, metodologi, sistematika, dan relevansi tugas akhir yang dikerjakan.

1.1 Latar Belakang

Teknologi yang semakin berkembang pesat, mendorong manusia untuk berinovasi agar memudahkan pekerjaan mereka. Khususnya untuk pekerjaan yang membosankan, berbahaya, di tempat yang kotor, maupun pekerjaan yang sulit untuk dilakukan. Salah satu inovasi dalam hal perkembangan teknologi ini adalah di bidang robotika. Penggunaan robot telah dikembangkan serta dilakukan berbagai *research* oleh manusia untuk diterapkan pada kehidupan sehari-hari manusia. Salah satu robot yang umum digunakan oleh manusia adalah jenis *mobile robot*. *Mobile robot* adalah suatu mesin otomatis yang bergerak (*mobile*) yang menjalankan pekerjaan sesuai fungsinya. Penggunaan *mobile robot* ini diharapkan dapat bekerja secara otomatis. Namun terdapat kendala yang akan dijumpai oleh *mobile robot* yaitu halangan (*obstacle*) untuk mencapai tujuan yang akan dilalui *mobile robot*. Maka dari itu dibutuhkan metode berdasarkan *input* yang dideteksi oleh sensor *mobile robot*.

Pada umumnya dalam perencanaan lintasan *mobile robot* harus ditentukan dahulu parameter –parameter dari *plant* yang akan dikendalikan. Maka dari itu, perlu adanya proses identifikasi dari *plant*. Tetapi berdasarkan kompleksitas dari *plant* parameter tersebut akan sulit untuk ditentukan. Sebuah pendekatan yang dilakukan dengan parameter dari *plant* yang sulit ditentukan dapat dilakukan dengan jaringan saraf tiruan. Metode RBF (*radial basis function*) adalah salah satu metode jaringan saraf tiruan dengan parameter yang tidak diketahui atau tanpa proses identifikasi model matematika. Namun melalui *differential drive* kinematik dan *inverse* kinematik.

Pada tugas akhir ini dilakukan perencanaan rute *mobile robot* agar *mobile robot* dapat menghindari obstacles dari posisi awal hingga posisi akhir yang telah ditentukan. *Mobile robot* yang digunakan adalah Quanser Qbot yang merupakan *autonomous ground robot* dari platform iRobot. Oleh karena itu, perencanaan rute *mobile robot* menggunakan aplikasi dari metode *radial basis function*

dengan kontroler *PID-fuzzy* agar diperoleh pergerakan *mobile robot* yang lebih *smooth*.

1.2 Permasalahan

Mobile robot akan bergerak dari posisi awal hingga menuju posisi akhir yang akan dituju. Permasalahan yang akan dibahas pada tugas akhir ini adalah perencanaan lintasan *mobile robot* menuju posisi akhir agar dapat menghindari *obstacles* dengan menggunakan metode *radial basis function*. Oleh karena itu dibuat suatu kontroler *PID – fuzzy* agar pergerakannya lebih *smooth* saat menghindari *obstacles*.

1.3 Tujuan

Tujuan dari penelitian tugas akhir ini yaitu memberi aksi kontrol untuk pergerakan dari *mobile robot* dalam mengenali keadaan lingkungan sekitar. Memberi aksi kontrol dengan metode *RBF* (*radial basis function*) agar lebih *smooth* saat menghindari *obstacle* atau rintangan yang terdapat pada lintasan yang akan dilaluinya.

1.4 Metodologi

Metodologi yang digunakan dalam penelitian tugas akhir ini adalah sebagai berikut :

1. Tinjauan Pustaka

Dilakukan dengan mengumpulkan materi serta informasi mengenai plant maupun metode yang akan digunakan melalui internet, paper dan sebagainya.

2. Pengambilan Data dan Identifikasi

Pada tahap ini dilakukan identifikasi plant dari *mobile robot* untuk menentukan parameter yang akan digunakan untuk melakukan desain pengendalian agar sesuai yang diinginkan.

3. Perancangan Kontroler

Pada tahap ini dilakukan desain kontroler *PID- fuzzy* dengan metode *RBF* (*radial basis function*) untuk system navigasi *mobile robot* dan dilakukan secara *real time*.

4. Implementasi dan Uji Coba

Kontroler yang sudah dirancang akan di implementasikan secara langsung pada plant berupa Qbot *Mobile robot* pada

rute yang sudah direncanakan lintasannya dan diberi *obstacles*.

5. Penulisan Buku Tugas Akhir

Penyusunan buku Tugas Akhir meliputi pendahuluan, dasar teori, perancangan sistem, implementasi, dan analisa serta penutup.

1.5 Sistematika

Pembahasan tugas akhir ini dibagi menjadi lima bab dengan sistematika sebagai berikut :

Bab 1 : Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, dan sistematika penulisan.

Bab 2 : Teori Penunjang

Bab ini membahas tinjauan pustaka untuk membantu penelitian, diantara teori Qbot mobile robot, teori identifikasi kontroler PID-fuzzy dan metode perencanaan lintasan dengan radial basis function.

Bab 3 : Perancangan Sistem

Pada bab ini dijelaskan mengenai perancangan lintasan yang akan dilalui mobile robot dengan menggunakan metode RBF serta identifikasi sistem mobile robot serta kontroler PID-fuzzy.

Bab 4 : Hasil dan Analisa

Bab ini memuat hasil implementasi kontroler dan pengujianya pada sistem.

Bab 5 : Penutup

Bab ini berisi kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

1.6 Relevansi

Hasil yang diperoleh dari tugas akhir ini diharapkan dapat diterapkan dan dikembangkan agar pemanfaatan fungsi dari *mobile robot* agar lebih optimal di masa mendatang.

(halaman ini sengaja dikosongkan)

BAB 2

TINJAUAN PUSTAKA

Pada Bab ini akan dibahas mengenai materi dasar dalam penyusunan Tugas Akhir. Beberapa hal yang dibahas meliputi tinjauan pustaka mengenai spesifikasi *mobile robot*, *Quanser Qbot*, *radial basis function*, dan teori kontroler *PID-fuzzy*.

2.1 Tinjauan Pustaka

Dengan berkebang pesatnya teknologi robot, maka semakin kompleks pula berbagai permasalahan yang akan dihadapi. Dalam hal ini khususnya *mobile robot*, yaitu robot yang bergerak atau *mobile* sesuai dengan fungsinya. Sebagian besar *mobile robot* dirancang untuk dapat beroperasi pada lingkungan kerja yang diinginkan, seperti robot permukaan, robot bawah air, dan robot udara. Perkembangan *mobile robot* menjadi peralatan canggih baik di bidang militer, kesehatan, pendidikan, lingkungan, SAR (*search and rescue*), bahkan cara penyajian di restaurant juga menggunakan *mobile robot*. *Mobile robot* yang dikembangkan saat ini dilengkapi dengan perangkat keras, kemampuan, kontrol dan aplikasi. Robot tersebut merupakan robot yang banyak dikembangkan untuk riset dan aplikasi termasuk pemetaan, pengintaian, operasi jarak jauh, dan navigasi otomatis.

Pada penelitian mengenai *mobile robot Qbot* sebelumnya yang dilakukan oleh Dwi Wulandari dalam Tugas Akhirnya telah dibahas mengenai *Path Planning mobile robot Qbot* dengan metode B-Spline untuk *path planning mobile robot* dan kontrol *Neuro Fuzzy* sebagai kontrol kecepatan *mobile robot*. Oleh karena itu, pada penelitian ini dilakukan perencanaan lintasan *mobile robot Qbot* dari titik awal menuju tujuan akhir agar mampu menghindari *obstacles* dengan menggunakan metode *Radial Basis Function* dan kontrol logika *PID Fuzzy* yang digunakan sebagai kontrol kecepatan *mobile robot Qbot*.

Perencanaan lintasan pada *mobile robot Qbot* dilakukan dengan mensimulasikan terlebih dahulu perencanaan lintasan dengan metode *Radial Basis Function* dengan mencari lintasan terpendek, lintasan digambarkan sebagai sekumpulan titik diatas bidang datar berskala 3x3 meter. Setiap titik dipresentasikan sesuai koordinat posisinya, yaitu x dan y. Sebelum melakukan perencanaan lintasan menggunakan metode RBF maka ditentukan terlebih dahulu posisi-posisi *obstacles*, titik awal *mobile robot*, serta titik tujuan *mobile robot* pada sumbu x dan y. Kemudian

ditentukan perkiraan lintasan yang akan dilewati mobile robot dengan masukan pada sumbu x (misalkan x_1, x_2, x_3 hingga x_n). Setelah itu baru melakukan penyusunan metode RBF serta menentukan layer layer nya. Untuk kontrol kecepatan pergerakan pada *mobile robot* Qbot digunakan kontroler logika *PID Fuzzy*.

2.2 Mobile Robot Qbot

Qbot *Mobile robot* merupakan sistem inovatif robot darat *autonomous* yang menggabungkan ilmu kendaraan darat dengan *Quanser Controller Module* (QCM). *Mobile robot* ini terdiri dari iRobot Create® *Robotic Platform*, sensor inframerah dan sonar serta sebuah kamera *Logitech Quickcam Pro 9000* USB seperti terlihat pada Gambar 2.1. QCM yang terpasang pada Qbot yang mana menggunakan komputer Gumstix untuk menjalankan QuaRC, perangkat lunak kontrol Quanser, dan Qbot kartu data akuisisi (DAC).



Gambar 2.1 Mobile Robot Qbot [3]

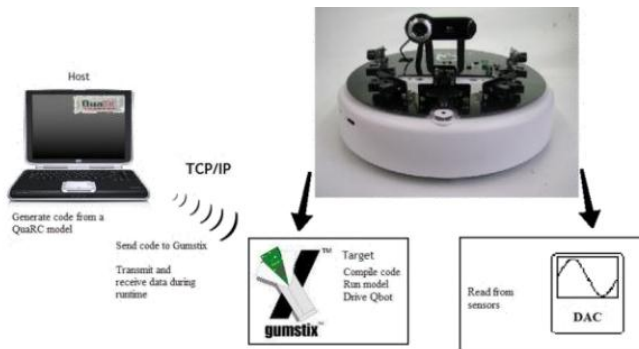
Berikut ini spesifikasi dan model parameter *mobile robot* Qbot dapat dilihat pada Tabel 2.1.

Tabel 2.1 Spesifikasi Model Mobile Robot Qbot

Simbol	Deskripsi	Value	Unit
D	Diameter Qbot <i>mobile robot</i>	0,34	m
H	Tinggi Qbot <i>mobile robot</i> (dengan tambahan kamera)	0,19	m
V_{max}	Kecepatan maksimum dari Qbot <i>mobile robot</i>	0,5	m/s

<i>M</i>	Berat total Qbot <i>mobile robot</i>	2,92	kg
----------	--------------------------------------	------	----

Antarmuka untuk QCM adalah MATLAB Simulink dengan QuaRC. Qbot dapat menerima perintah, melalui tiga blok yang berbeda, yaitu blok Roomba untuk menggerakkan Qbot, blok HIL untuk membaca dari sensor dan/atau menulis untuk *output* servo dan blok OpenCV untuk mengakses kamera. Kontroler dikembangkan pada Simulink dengan QuaRC dan model ini diunduh dan disusun pada target (Gumstix). Diagram dari konfigurasi ini ditampilkan dalam Gambar 2.2.



Gambar 2.2 Hierarki Komunikasi *Mobile Robot Qbot* [3]

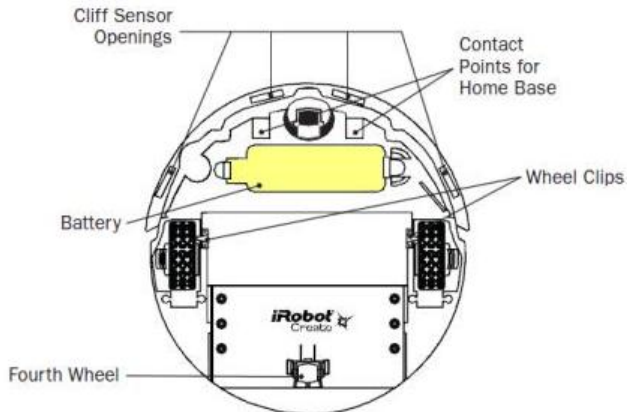
2.2.1 Baterai *Mobile Robot Qbot*

Qbot ini didukung oleh baterai *Advance Power System* (APS) yang disediakan oleh iRobot seperti pada Gambar 2.7. Baterai ini terletak di bawah Qbot dan dapat berlangsung terus menerus selama sekitar dua jam setelah terisi penuh.



Gambar 2.3Baterai *Mobile Robot Qbot* [3]

Baterai terletak dibagian bawah seperti pada Gambar 2.8. Lampu daya Qbot mengindikasikan tingkat daya baterai. Lampu hijau menandakan baterai masih terisi penuh, kemudian secara bertahap berubah menjadi merah bila baterai akan habis. Baterai memerlukan waktu kurang dari tiga jam untuk pengisian baterai. Ketika pengisian, lampu daya akan berkedip perlahan dengan warna jingga dan akan berubah menjadi hijau ketika terisi penuh.



Gambar 2.4 Letak Baterai pada *Mobile Robot Qbot* [3]

2.2.2 Sensor Inframerah dan Sonar

SHARP 2Y0A02 seperti pada Gambar 2.9 merupakan sensor inframerah dengan jarak 20-150 cm. Terdapat lima buah sensor inframerah yang terpasang pada Qbot. Sensor terhubung ke saluran *input analog* dari Qbot DAC, yang kemudian dapat dibaca dengan menggunakan blok HIL *Read Write*.



Gambar 2.5 Sensor Inframerah SHARP 2Y0A02 [3]

Sensor sonar MaxSonar-EZ0 mendeteksi objek dari 0 inci sampai 254 inci dengan resolusi 1 inci. Objek antara 0 inci dan 6 inci berkisar sebagai 6 inci. Terdapat tiga sensor pada Qbot. Sensor yang terhubung ke saluran masukan lain dari Qbot DAC, yang kemudian dapat dibaca dengan menggunakan blok HIL *Read Write*. Gambar 2.10 menunjukkan sensor sonar MaxSonar-EZ0.



Gambar 2.6 Sensor Sonar MaxSonar-EZ0 [3]

2.3 QUARC

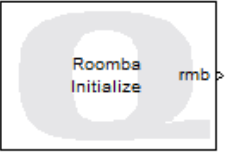
QUARC (*Quanser Real-Time Control*) adalah perangkat lunak yang menyediakan fasilitas untuk mengembangkan dan menguji coba rancangan kontroler pada komputer *host* dengan menggunakan perangkat lunak Simulink MATLAB, dengan model yang dirancang dapat langsung diunduh dan dieksekusi pada Gumstix dengan menggunakan komunikasi *wireless* secara *real time*. Pada waktu yang bersamaan, operator dapat memantau dan mengukur nilai-nilai dari sensor dan mengatur parameter-parameter kontroler langsung dari komputer *host*. Pada MATLAB Simulink terdapat beberapa blok yang digunakan untuk melakukan komunikasi dengan *mobile robot* Qbot seperti yang terdapat pada Tabel 2.2 dan Tabel 2.3.

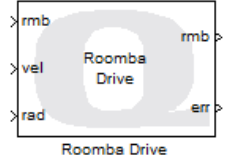
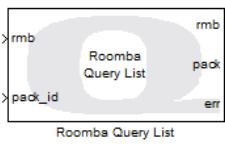
Tabel 2.2 Deskripsi Blok Set untuk *Mobile Robot* Qbot


Blok Set	Deskripsi
Interface	Blok set ini mengimplementasikan dasar <i>Aplication Program Interfaces</i> (API) yang

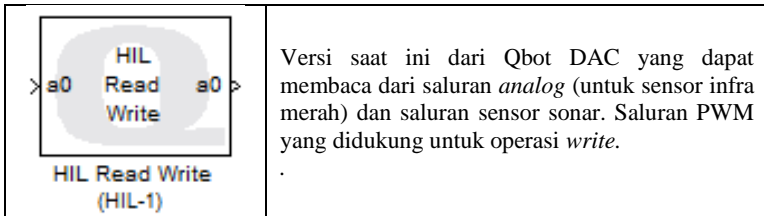
	<p>disediakan oleh <i>iRobot Create</i>®. API dapat dikategorikan berdasarkan fungsional berikut:</p> <ul style="list-style-type: none"> • Sambungkan konfigurasi serial antara kontroler tingkat tinggi (misalnya, PC atau Gumstix) dan Qbot. • Pengaturan <i>mode</i> operasi Qbot • Mengakses informasi sensorik Qbot • Konfigurasi <i>hardware</i> lainnya yang sesuai (misalnya, pengaturan <i>port I/O digital</i> dan <i>analog</i>, dan mengubah warna LED)
Aplikasi	Blok set ini memungkinkan pengguna untuk menerapkan algoritma navigasi menggunakan informasi sensorik yang tersedia. Blok ini hanya menggunakan data <i>encoder</i> roda dan <i>bump sensors</i> untuk navigasi dan menghindari rintangan.
<i>Image Processing</i>	Blok set ini mengimplementasikan akuisisi citra dan pengolahan fungsi menggunakan <i>Library Open CV</i> . Blok set pengolahan citra memungkinkan untuk menangkap gambar dari kamera USB, untuk proses <i>online</i> dan menyimpannya ke <i>disk</i> untuk analisis lebih lanjut.

Tabel 2.3 Deskripsi Blok QUARC pada *Mobile Robot Qbot*

Blok	Deskripsi
	<p>Blok ini diperlukan untuk membuat sambungan serial ke Qbot. Qbot diidentifikasi oleh <i>Universal Resource Identifier</i> (URI), seperti <i>serial://localhost:1</i> dengan parameter <i>baud=57600, word=8, parity=none, stop=1</i>, di mana <i>port</i> komunikasi 1 dari target terhubung ke <i>port serial</i> Qbot dengan parameter yang ditentukan. Catatan: <i>Output</i> “rmb” dari blok ini harus terhubung ke <i>input</i> “rmb” dari blok di blok Roomba yang ditetapkan dalam rangka untuk mengakses Roomba.</p>

 <p>Roomba Drive</p>	<p>Blok ini menggerakkan Qbot dengan dua <i>input</i>, yaitu kecepatan dan radius. Untuk menggerakkan lurus, <i>input</i> radius mengambil 32768 atau 32767.</p>
 <p>Roomba Query List</p>	<p>Blok ini mengambil berbagai data <i>sensor</i> dari Qbot. <i>Input pack_id</i> berkisar 7-42, dan setiap <i>input</i> akan menampilkan nilai dari <i>output pack</i>. Lihat halaman Help MATLAB untuk deskripsi lengkap untuk masing-masing paket ID.</p> <p>Nilai <i>pack_id</i> yang penting :</p> <ul style="list-style-type: none"> 8 = Dinding (0 = tidak ada dinding, 1 = dinding terlihat) 19 = Jarak (Jarak yang Qbot telah melakukan perjalanan dalam milimeter) 20 = Sudut (Sudut dalam derajat Qbot berubah) 22 = Tegangan (Tegangan dari baterai Qbot dalam milivolt) 23 = Arus (Arus dalam mili ampere mengalir atau keluar dari baterai Qbot)

Blok Set	Deskripsi
 <p>HIL Initialize HIL-1 (q8-0)</p>	<p>Blok inialisasi yang diperlukan dalam semua model QuaRC menggunakan HIL Card. Hanya satu blok HIL <i>Initialize</i> yang dibutuhkan dalam model untuk mengatur satu kendaraan. Blok HIL <i>Read Write</i> harus referensi blok ini untuk menentukan <i>board</i> yang digunakan.</p>

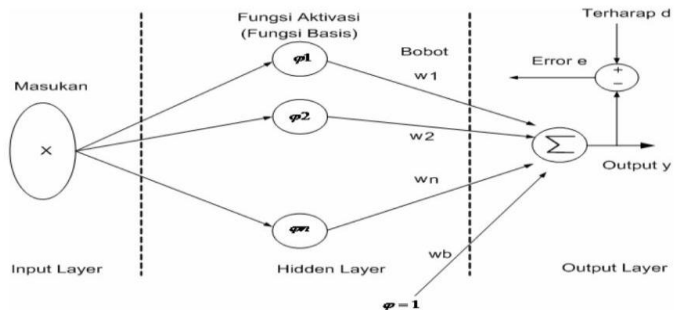


2.4 Radial Basis Function

Radial Basis Function Network merupakan sebuah metode yang mempunyai struktur neural network yang terdiri dari 3 layer, yang meliputi [4][5]:

- Input nodes
- Hidden neuron layer, yang berguna sebagai transformasi non-linier dari input melalui RBF
- Output layer, yang merupakan kombinasi linier dari output hidden layer

Struktur dari Radial Basis Function Neural Network diilustrasikan seperti dalam Gambar



Gambar 2.7 Struktur RBF Network

Ketika RBF Network diaplikasikan untuk menyelesaikan permasalahan yang kompleks, maka permasalahan tersebut pada dasarnya diselesaikan dengan langkah awal yaitu mentransformasikan permasalahan tersebut ke dalam high dimensional space dalam kontrol non-linier, kemudian memisahkannya ke dalam kelas-kelas tertentu dalam output layer. Pemisahan kelas-kelas tersebut diperjelas dalam

Teorema Cover yang berbunyi bahwa “Masalah pola klasifikasi kompleks yang ada dalam ruang dimensi tinggi secara non-linear, lebih cenderung linear jika dipisah satu sama lain dalam ruang dimensi yang rendah”.

Dari Teorema Cover tersebut, kita dapat menyatakan bahwa mapping non-linear digunakan untuk mentransformasikan permasalahan klasifikasi pemisahan non-linear ke dalam pemisahan linier dengan probabilitas yang tinggi. Neural network didesain untuk melaksanakan non-linear mapping dari input ke hidden space, kemudian diikuti oleh linear mapping dari hidden space ke output space. Persamaan dari Radial Basis Function untuk interpolasi adalah sebagai berikut, seperti dijabarkan pada persamaan (2.2):

$$F(x) = \sum_{i=1}^N w_i \phi(|x - C_i|), \quad (2.1)$$

$$\Phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (2.2)$$

Di mana :

Φ adalah kumpulan dari N fungsi non linear, disebut sebagai radial basis. Persamaan 2.2 adalah fungsi Gaussian.

w_i adalah merepresentasikan bobot antara neuron 1 dari hidden layer dan output layer

$\| \cdot \|$ menunjukkan *euclidean norm*

C_i adalah titik pusat dari masing – masing fungsi Gaussian, dimana $C_i \in \mathbb{R}^2$, $i = 1, 2, 3, \dots, N$

2.5 Kontroler PID Fuzzy

2.5.1 Logika Fuzzy

Logika Fuzzy adalah salah satu bentuk pendekatan logika manusia, dimana representasi suatu kejadian didistribusikan ke dalam sejumlah istilah bahasa (yang menyatakan level kualitatif) dengan nilai kebenaran tiap level terletak antara 0 sampai dengan 1.

Struktur dasar dari logika fuzzy meliputi himpunan fuzzy, himpunan pendukung, dan fungsi keanggotaan (*membership function*). Himpunan fuzzy adalah suatu himpunan yang beranggotakan sejumlah istilah dalam pengertian bahasa yang menyatakan level kualitatif dari semesta pembicaraan x . Contoh adalah pengukuran kecepatan motor dapat didistribusikan ke dalam beberapa istilah bahasa yang menyatakan level kualitatif dari kecepatan putaran motor. Himpunan pendukung adalah anggota dari himpunan fuzzy yang berupa istilah bahasa yang menyatakan himpunan bagian dari semesta pembicaraan x . Fungsi

keanggotaan (*membership function*) adalah suatu fungsi yang didefinisikan suatu anggota himpunan fuzzy yang menggambarkan derajat kebenaran suatu kejadian dalam semesta pembicaraan x , dinyatakan dalam tingkat keanggotaan (*grade of membership*) dengan nilai antara 0 sampai dengan 1.

Operasi dari logika fuzzy meliputi fuzzifikasi, menentukan rule base, inference, dan defuzzifikasi. Fuzzifikasi adalah transformasi nilai real menjadi nilai suatu himpunan fuzzy yang dinyatakan dalam tingkat keanggotaan tiap anggota himpunan pendukung melalui fungsi keanggotaan yang didefinisikan. Menentukan rule base adalah memilih penentuan atau logika dari himpunan pendukung, jika dimisalkan e adalah *error* dari kecepatan motor, dan u adalah sinyal control maka jika *error* (e) positif besar maka sinyal kontrol (u) besar, jika e nol maka u nominal dan seterusnya. Inference merupakan pengambilan data yang berasal dari rule base serta nilai yang dihitung dari fuzzifikasi, jika dimisalkan u_1, u_2, u_3 adalah sinyal control 1, 2, 3 maka dicari operasi OR (minimal) dari u_1, u_2, u_3 . Jika terdapat beberapa nilai u_1 maka dengan operasi AND (maksimal) dapat dicari nilainya. Defuzzifikasi merupakan kebalikan dari proses fuzzifikasi, yaitu mentransformasikan harga fuzzy ke harga singular bukan fuzzy. Secara umum proses defuzzifikasi ini dinyatakan sebagai $u_0 = \text{defuzzier}(u)$. Atau dengan persamaan 2.3 dibawah ini :

$$U(k) = \frac{u_1C_1 + u_2C_2 + \dots + u_nC_n}{u_1 + u_2 + \dots + u_n} \quad (2.3)$$

Dimana u adalah nilai dari inference u_1 sampai dengan u_n .

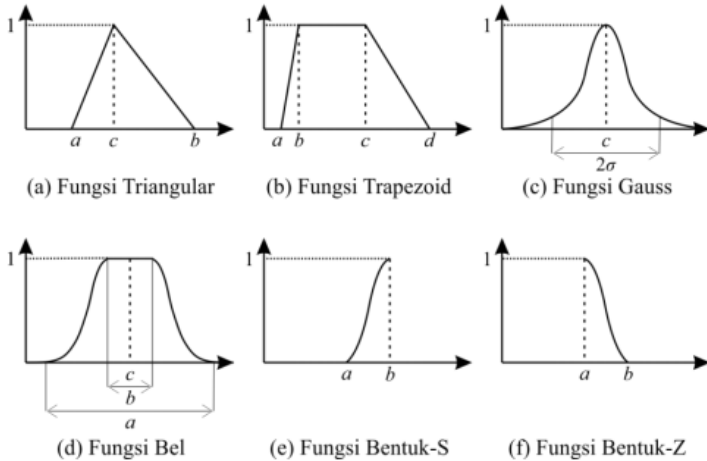
C adalah nilai dari output fuzzifikasi.

2.5.1.1 Himpunan fuzzy

Himpunan *Fuzzy* merupakan suatu himpunan yang beranggotakan sejumlah istilah dalam pengertian bahasa yang menyatakan level kualitatif dari semesta pembicaraan. Misalnya, pengukuran kecepatan putaran motor dapat diterjemahkan ke dalam beberapa istilah bahasa yang menyatakan level kualitatif dari kecepatan putaran motor sehingga apabila semesta pembicaraan berupa kecepatan putaran motor, maka dapat dibuat suatu himpunan *Fuzzy* yaitu “Sangat Lambat”, “Lambat”, “Sedang”, “Cepat”, “Sangat Cepat”.

2.5.1.2 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) merupakan kurva yang menunjukkan pemetaan *input* data ke dalam nilai keanggotaannya dengan rentang nol sampai satu. Gambar 2.8 menunjukkan bentuk-bentuk dari fungsi keanggotaan.



Gambar 2.8 Bentuk-bentuk *Membership Function*

Berikut beberapa fungsi keanggotaan yang umum digunakan, yaitu :

a. Fungsi Segitiga

Fungsi segitiga merupakan gabungan antara dua fungsi linear. Fungsi ini ditentukan oleh tiga parameter $\{a, b, c\}$ dengan ketentuan $a < c < b$. Pada Gambar 2.8 ditampilkan bentuk fungsi segitiga. Fungsi keanggotaan fungsi segitiga, yaitu :

$$\mu(x) = \begin{cases} 0; & x \leq a \\ \frac{x-a}{c-a}; & a \leq x \leq c \\ \frac{b-x}{b-c}; & c \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.4)$$

b. Fungsi Trapesium

Fungsi trapesium pada dasarnya seperti fungsi segitiga yang memiliki beberapa titik dengan nilai keanggotaan satu. Fungsi ini ditentukan oleh empat parameter $\{a, b, c, d\}$ dengan ketentuan $a < c \leq d < b$. Pada Gambar 2.8 ditampilkan bentuk fungsi trapesium. Fungsi keanggotaan fungsi trapesium, yaitu :

$$\mu(x) = \begin{cases} 0; & x \leq a \\ \frac{x-a}{c-a}; & a \leq x \leq c \\ 1; & c \leq x \leq d \\ \frac{b-x}{b-d}; & d \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.5)$$

c. Fungsi S

Fungsi S didefinisikan dengan menggunakan tiga parameter, yaitu nilai keanggotaan nol (α), nilai keanggotaan penuh (γ), dan titik infleksi (β). Titik infleksi yaitu titik yang memiliki domain 50% benar. Pada Gambar 2.8 ditampilkan bentuk fungsi S. Fungsi keanggotaan fungsi S, yaitu :

$$\mu(x) = \begin{cases} 0; & x \leq \alpha \\ 2 \left(\frac{x-\alpha}{\gamma-\alpha} \right)^2; & \alpha \leq x \leq \beta \\ 1 - 2 \left(\frac{x-\alpha}{\gamma-\alpha} \right)^2; & \beta \leq x \leq \gamma \\ 1; & x \geq \gamma \end{cases} \quad (2.6)$$

d. Fungsi Phi (π)

Fungsi phi berbentuk lonceng, menggunakan dua parameter yaitu (γ) untuk menunjukkan derajat keanggotaan bernilai satu yang terletak di pusat himpunan, dan lebar fungsi (β). Pada Gambar 2.8 ditampilkan bentuk fungsi phi. Fungsi keanggotaan fungsi phi, yaitu :

$$\mu(x) = \begin{cases} S \left(x; \gamma - \beta, \gamma - \frac{\beta}{2}, \gamma \right); & x \leq \gamma \\ 1 - S \left(x; \gamma + \beta, \gamma + \frac{\beta}{2}, \gamma \right); & x > \gamma \end{cases} \quad (2.7)$$

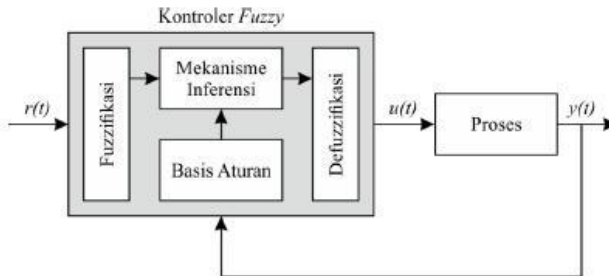
e. Fungsi *Gaussian*

Fungsi *gaussian* menggunakan dua parameter, yaitu (c) untuk menunjukkan nilai domain pada pusat kurva, dan (σ) menunjukkan lebar fungsi. Pada Gambar 2.8 ditampilkan bentuk fungsi *gaussian*. Fungsi keanggotaan fungsi *gaussian*, yaitu :

$$\mu(x) = e^{-\left(\frac{x-c}{\sigma}\right)^2} \quad (2.8)$$

2.5.1.3 Kontroler Logika Fuzzy

Kontroler logika *Fuzzy* merupakan suatu kontroler yang proses perhitungan sinyal kontrolnya melalui operasi himpunan *Fuzzy* meliputi proses Fuzzifikasi, relasi *Fuzzy*, inferensi *Fuzzy* serta Defuzzifikasi seperti terlihat pada Gambar 2.9.

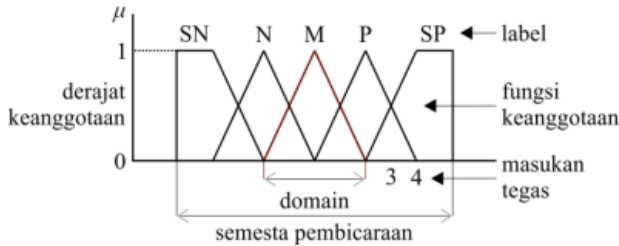


Gambar 2.9 Struktur Kontroler *Fuzzy* [4]

Proses di dalam himpunan *Fuzzy* pada kontroler *Fuzzy* antara lain sebagai berikut :

a. **Fuzzifikasi**

Fuzzifikasi merupakan proses transformasi nilai *real* (nilai bukan *Fuzzy*) menjadi nilai suatu himpunan *Fuzzy* yang dinyatakan dalam derajat keanggotaan yang didefinisikan. Struktur fungsi keanggotaan *Fuzzy* dapat dilihat pada Gambar 2.10. Derajat keanggotaan adalah derajat dari masukkan tegas pada sebuah fungsi keanggotaan, memiliki nilai 0 sampai 1. Fungsi keanggotaan mendefinisikan nilai *Fuzzy* dengan melakukan pemetaan nilai tegas berdasarkan daerahnya untuk diasosiasikan dengan derajat keanggotaan.



Gambar 2.10 Struktur Fungsi Keanggotaan Fuzzy

Masukkan tegas pada umumnya merupakan hasil pengukuran parameter eksternal dari sistem kontrol. Label merupakan deskripsi nama untuk menunjukkan suatu fungsi keanggotaan Fuzzy yang memiliki domain (lebar fungsi keanggotaan Fuzzy) tertentu. Semesta pembicaraan memiliki jarak yang mencakup seluruh masukan tegas yang mungkin ada. Bentuk fungsi keanggotaan harus mewaliki variabel masukan tegas, namun bentuk yang digunakan dibatasi oleh kemampuan *tool* dalam melakukan perhitungan. Bentuk fungsi yang rumit membutuhkan persamaan fungsi yang lebih kompleks.

b. Basis Aturan

Basis aturan merupakan deskripsi linguistik terhadap variabel *input* dan *output*. Penentuan basis aturan yang dipakai dalam mengontrol suatu *plant* dapat melalui metode heuristik maupun deterministik. Metode heuristik didasarkan pada pengetahuan terhadap *plant* dan perilaku dari *plant* yang akan dikontrol. Sedangkan metode deterministik diperoleh melalui identifikasi struktur dan parameter dari aturan kontrol. Pemetaan *input* dan *output* pada sistem Fuzzy direpresentasikan dalam pernyataan **IF** *premis* **THEN** *konsekuensi*. Pada kontroler Fuzzy, *input* kontroler direpresentasikan dalam *premis* dan *output* kontroler direpresentasikan dalam *konsekuensi*. Jumlah basis aturan dari suatu sistem Fuzzy ditentukan dari jumlah variabel pada *input* dan jumlah *membership function* pada variabel masukan, dirumuskan dalam Persamaan 2.6.

$$\prod_{i=1}^n N_i = N_1 \times N_2 \times \dots \times N_n \quad (2.9)$$

Dimana N_i merupakan jumlah *membership function* pada variabel *input* i . Sebagai contoh, apabila variabel *input* pertama memiliki tiga *membership function* dan variabel *input* kedua memiliki tiga *membership function*, maka jumlah basis aturan adalah $3 \times 3 = 9$ aturan.

c. Mekanisme Inferensi

Mekanisme inferensi pada kontroler *Fuzzy* merupakan mekanisme operasi matematika yang dilakukan sesuai dengan *premis* yang diberikan. Pada basis aturan terdapat dua operasi matematika yang dapat dilakukan yaitu *AND* dan *OR*. Terdapat beberapa tipe mekanisme inferensi *Fuzzy*, antara lain Mamdani, Larsent dan Takagi sugeno. Perbedaan dari metode ini terletak pada pengambilan kesimpulan logika *Fuzzy*. Pada metode Mamdani maupun Larsent, kesimpulan logika *Fuzzy* berupa derajat keanggotaan sehingga dalam menyimpulkan suatu logika *Fuzzy* dibutuhkan proses defuzzifikasi. Sedangkan pada tipe Takagi Sugeno, kesimpulan logika *Fuzzy* berupa suatu persamaan sehingga tidak diperlukan proses defuzzifikasi. Kelebihan pada logika *Fuzzy* tipe Mamdani dan Larsent lebih sederhana, akan tetapi diperlukan kemampuan untuk mengetahui karakteristik *plant* untuk menentukan batasan keluaran kontroler. Pada tipe Takagi-Sugeno tidak diperlukan pengetahuan mengenai karakteristik dari *plant* akan tetapi diperlukan perhitungan yang lebih rumit untuk persamaan pada bagian *konsekuen*.

d. Defuzzifikasi

Defuzzifikasi adalah suatu proses mentransformasikan harga fuzzifikasi yang telah diinferensi kedalam harga bukan *Fuzzy* atau harga aktual. Beberapa metode dalam proses defuzzifikasi, yaitu *Center of Area*, *Mean of Maxima*, *Weighted Sum* dan lainnya.

1. Center of Area

Metode *Center of Area* digunakan untuk menentukan nilai titik tengah area yang merupakan titik pusat massa dari kombinasi fungsi-fungsi keanggotaan. Secara umum,

persamaan untuk metode *Center of Area* ditunjukkan dengan Persamaan 2.7.

$$U_0 = \frac{\sum_{k=1}^m u_k(T) \cdot \mu_u(u_k(T))}{\sum_{k=1}^m \mu_k(u_k(T))}, \forall u \in U(T) \quad (2.10)$$

2. Mean of Maxima

Metode *Mean of Maxima* mengambil semua nilai tiap fungsi keanggotaan dengan derajat keanggotaan maksimum dan menghitung rata-rata dari nilai-nilai tersebut sebagai keluaran tegas. Persamaan 2.8 menunjukkan persamaan umum metode *Mean of Maxima*.

$$U_0 = \frac{\sum_n \max(\mu_A^n) \cdot y_n}{\sum_n \max(\mu_A^n)} \quad (2.11)$$

3. Weighted Sum

Metode *Weighted Sum* merupakan metode defuzzifikasi dengan fungsi keanggotaan berupa fungsi *singleton*. Persamaan 2.9 menunjukkan persamaan defuzzifikasi fungsi *singleton*.

$$U_0 = \frac{\sum_i \mu_A^i s_i}{\sum_i \mu_A^i} \quad (2.12)$$

2.5.2 PID (Proportional Integrative Derivative)

Kontroler Proporsional ditambah Integral ditambah Differensial (PID) merupakan kontroler yang aksi kontrolernya mempunyai sifat proporsional, integral dan differensial terhadap sinyal kesalahan.

Dengan menentukan nilai penguatan proporsional K_p , waktu integral τ_i dan waktu differensial τ_d yang tepat diharapkan respon plant orde kedua tanpa delay sesuai dengan spesifikasi performansi yang diinginkan.

Komponen sistem jika plant orde kedua tanpa delay seperti persamaan 2.13 berikut.

$$\frac{C(s)}{U(s)} = \frac{K}{\frac{1}{W_n^2} s^2 + \frac{2\xi}{W_n} s + 1} \quad (2.13)$$

Dimana :

K = gain overeall,
 W_n = frekuensi alami tak teredam,
 ξ = rasio peredam

Sedangkan control PID adalah seperti persamaan 2.14 berikut.

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2.14)$$

Dimana :

K_p = penguat integral,

τ_i = waktu integral,

τ_d = waktu diferensial.

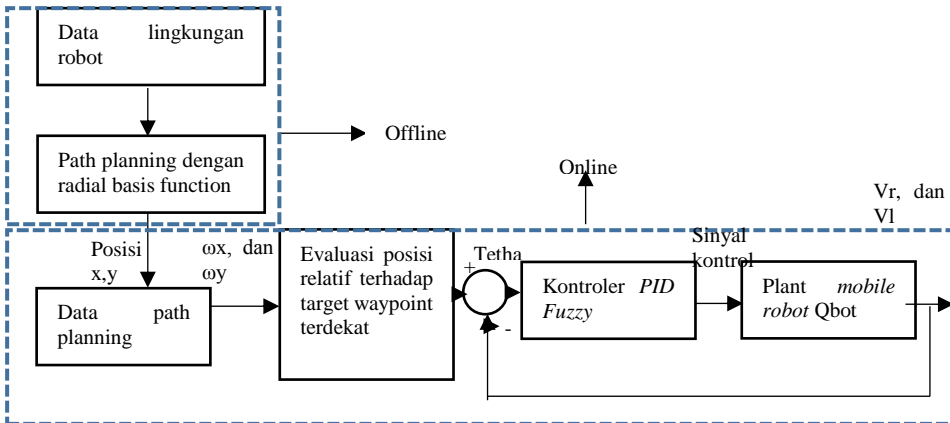
(halaman ini sengaja dikosongkan)

BAB 3 PERANCANGAN SISTEM

Pada bab ini dibahas mengenai perancangan sistem *path planning* serta perancangan untuk memberikan aksi kontrol pada *mobile robot* Qbot, yaitu perancangan diagram blok sistem, model matematika *mobile robot* Qbot, perancangan *path planning* dengan *radial basis function*, perancangan kontroler *PID Fuzzy*, dan perancangan *software*.

3.1 Diagram Blok Sistem

Diagram blok pada gambar 3.1 menunjukkan keseluruhan dari sistem perencanaan lintasan untuk *mobile robot* Qbot menggunakan metode *radial basis function* dan kontroler *PID Fuzzy*.



Gambar 3.1 Diagram Blok Fungsional

Pada gambar 3.1 diagram blok fungsional sistem menunjukkan bahwa sistem terdiri dari dua sub sistem, yaitu offline dan online. Pada sub sistem offline terdiri dari perancangan serta pemrograman untuk memperoleh lintasan yang sudah ditentukan serta lintasan yang dihasilkan oleh radial basis function. Pada sub sistem online merupakan hasil data dari lintasan yang dihasilkan oleh radial basis function, yaitu berupa titik posisi x dan y yang akan dievaluasi posisi pada *trajectory* agar menghasilkan ω_x dan ω_y . Kemudian dari ω_x dan ω_y dihasilkan sudut

tetha yang akan digunakan oleh *plant mobile robot* menentukan V_r dan V_l .

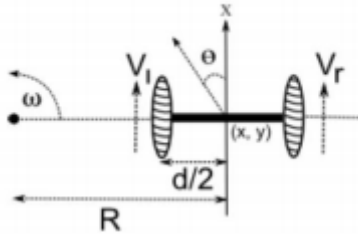
3.2 Model Matematika *Mobile Robot Qbot*

Berikut ini catatan yang diadopsi dari Dudek dan Jenkin mengenai prinsip komputasi mobile robot. Langkah pertama dalam menganalisis sistem kontrol adalah menurunkan model matematika sistem. Setelah model diperoleh, digunakan metode untuk analisis reformasi sistem. Pada penelitian Tugas Akhir ini, model matematika untuk Qbot mobile robot menggunakan differential drive kinematics dan invers kinematic.

3.2.1 Differential Drive Kinematics

Notasi dan model matematika untuk kinematik dari mobile robot beroda akan dirancang dalam sub bab ini berdasarkan referensi yang ada. Robot beroda mempunyai beberapa keistimewaan sederhana yang membuat robot ini mudah untuk dimodelkan, karena beroperasi pada bidang dua dimensi. Secara khusus, robot ini memiliki dua roda koaksial. Perbedaan kecepatan antara dua roda menyebabkan rotasi robot sekitar pusat sumbu, sementara roda menghasilkan gerakan di depan atau arah sebaliknya. Hal ini diketahui bahwa dinamika pada kendaraan nyata, pada kecepatan tinggi menyebabkan gerakan vertikal yang dikompensasikan dengan menerapkan suspensi untuk desain mereka. Namun, karena robot ini beroperasi sangat rendah, kecepatan gerak vertikal hampir diabaikan dan tidak ada suspensi yang diperlukan.

Sebuah differential drive mobile robot terdiri dari dua roda yang terpasang pada aksis yang sama, dan tiap roda bisa dikontrol secara bebas untuk menggerakkan robot ke arah depan atau ke belakang. Ketika kecepatan tiap roda dapat bervariasi untuk meraih gerakan melingkar, robot harus memutar sekitar titik yang disebut sebagai ICC (Instantaneous Center of Curvature), seperti yang ditunjukkan pada Gambar 3.2.



Gambar 3.2 Kinematik dari Differential Robot [6]

Lintasan robot dapat dikendalikan dengan cara mengatur kecepatan pada tiap rodanya. Kecepatan rotasi ω sekitar ICC harus sama pada kedua roda. Maka, persamaan berikut ini menyusun hubungan antara parameter gerak dari differential drive mobile robot.

$$\omega \left(R + \frac{d}{2} \right) = V_{right} \quad (3.1)$$

$$\omega \left(R - \frac{d}{2} \right) = V_{left} \quad (3.2)$$

Dimana d adalah jarak antara titik pusat dari kedua roda, V_{right} dan V_{left} adalah kecepatan roda kanan dan kiri saat menyusuri lantai, dan R adalah jarak dari ICC ke titik tengah antar roda.

Persamaan 3.1 dan 3.2 dapat diselesaikan pada setiap kasus waktu untuk R dan ω sebagai berikut. Mencari persamaan R adalah.

$$V_{right} = \omega \left(R + \frac{d}{2} \right) \quad \text{maka} \quad \omega = \frac{V_{right}}{\left(R + \frac{d}{2} \right)} \quad (3.3)$$

$$V_{left} = \omega \left(R - \frac{d}{2} \right) \quad \text{maka} \quad \omega = \frac{V_{left}}{\left(R - \frac{d}{2} \right)} \quad (3.4)$$

$$\text{Karena } \omega = \omega \text{ maka} \quad \frac{V_{right}}{\left(R + \frac{d}{2} \right)} = \frac{V_{left}}{\left(R - \frac{d}{2} \right)} \quad (3.5)$$

$$V_{right} \left(R - \frac{d}{2} \right) = V_{left} \left(R + \frac{d}{2} \right) \quad (3.6)$$

$$V_{right} R - \frac{V_{right} d}{2} = V_{left} R + \frac{V_{left} d}{2} \quad (3.7)$$

$$V_{right} R - V_{left} R = \frac{V_{right} d}{2} + \frac{V_{left} d}{2} \quad (3.8)$$

$$R(V_{right} - V_{left}) = \frac{d}{2}(V_{right} + V_{left}) \quad (3.9)$$

$$R = \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} \quad (3.10)$$

Selanjutnya mencari persamaan ω ,

$$V_{right} = \omega \left(R + \frac{d}{2} \right) \text{ maka } \omega R = V_{right} - \omega \frac{d}{2} \quad (3.11)$$

$$V_{left} = \omega \left(R - \frac{d}{2} \right) \text{ maka } \omega R = V_{left} + \omega \frac{d}{2} \quad (3.12)$$

$$V_{right} - \omega \frac{d}{2} = V_{left} + \omega \frac{d}{2} \quad (3.13)$$

$$2 \omega \frac{d}{2} = V_{right} - V_{left} \quad (3.14)$$

$$d \omega = V_{right} - V_{left} \quad (3.15)$$

$$\omega = \frac{V_{right} - V_{left}}{d} \quad (3.16)$$

Dari Persamaan 3.10 dan Persamaan 3.16 dapat diketahui hubungan antara R dan ω untuk mencari kecepatan linier (V).

$$V = \omega R = \frac{V_{right} - V_{left}}{d} \times \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} = \frac{V_{right} - V_{left}}{2} \quad (3.17)$$

3.2.2 Lokasi ICC

Pada Gambar 3.2 diasumsikan bahwa robot berada di sebuah posisi (x, y) dan menuju ke arah yang membuat sudut θ dengan X aksis. Dengan mengetahui kecepatan V_{right} , V_{left} dan menggunakan Persamaan 3.10 dan Persamaan 3.16, lokasi ICC (ICC_x , ICC_y) dapat ditentukan pada Persamaan 3.18 dan Persamaan 3.19.

$$ICC_x = x - R \sin \theta = x - \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} \sin \theta \quad (3.18)$$

$$ICC_y = y + R \cos \theta = y + \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} \cos \theta \quad (3.19)$$

3.2.3 Forward Kinematik

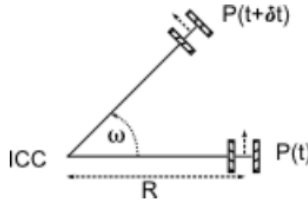
Berikut ini penjelasan permasalahan forward kinematics tentang bagaimana kecepatan yang diberikan pada roda dan konfigurasi posisi awal robot $(x, y, \theta)_{t=0}$ dapat menentukan posisi robot (x, y, θ) .

3.2.3.1 Pose Relative Robot ke Lokasi ICC

Diberikan sebuah kecepatan yang tidak bervariasi pada V_{right} dan V_{left} , lokasi ICC (ICC_x , ICC_y) akan menjadi posisi tetap. Karena itu, pada waktu $t+\delta t$ pose robot akan menjadi:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \times \delta t) & -\sin(\omega \times \delta t) & 0 \\ \sin(\omega \times \delta t) & \cos(\omega \times \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega \times \delta t \end{bmatrix}$$

Dimana (x, y, θ) dan (x', y', θ') adalah posisi robot pada waktu t dan $t+\delta t$, berturut-turut. Persamaan ini menggambarkan gerak berputar robot sekitar ICC nya dengan sebuah radius dari lengkungan R dan sebuah kecepatan sudut ω , seperti pada Gambar 3.3.



Gambar 3.3 Forward Kinematics Relative to ICC

3.2.3.2 Pose Relative Robot pada Posisi Awal Robot

Persamaan gerak umum robot yang mampu bergerak dalam arah $\theta(t)$ tertentu pada kecepatan $V(t)$ yang diberikan, dijelaskan sebagai berikut,

$$\begin{aligned} x(t) &= \int_0^t V(t) \cos[\theta(t)] dt \\ y(t) &= \int_0^t V(t) \sin[\theta(t)] dt \\ \theta(t) &= \int_0^t \omega(t) dt \end{aligned} \quad (3.20)$$

Untuk sebuah robot differential drive seperti Qbot, Persamaan 3.20 menjadi,

$$\begin{aligned} x(t) &= \frac{1}{2} \int_0^t [V_r(t) + V_l(t)] \cos[\theta(t)] dt \\ y(t) &= \frac{1}{2} \int_0^t V(t) \sin[\theta(t)] dt \end{aligned} \quad (3.21)$$

$$\theta(t) = \frac{1}{d} \int_0^t \omega(t) dt$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V \cos(\theta) \delta t \\ y + V \sin(\theta) \delta t \\ \theta + \omega \delta t \end{bmatrix} \quad (3.22)$$

Demikian pula pada Persamaan 3.21,

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + \frac{1}{2} [V_r + V_l] \cos(\theta) \delta t \\ y + \frac{1}{2} [V_r + V_l] \sin(\theta) \delta t \\ \theta + \frac{1}{d} [V_r - V_l] \delta t \end{bmatrix} \quad (3.23)$$

Dengan demikian, untuk kasus khusus dari $V_{left} = V_{right} = V$ dan $V_{left} = V_{right} = V$, Persamaan 3.24 dan Persamaan 3.25 menjadi Persamaan 3.22 dan Persamaan 3.23.

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V \cos(\theta) \delta t \\ y + V \sin(\theta) \delta t \\ \theta \end{bmatrix} \quad (3.24)$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + \frac{2V \delta t}{d} \end{bmatrix} \quad (3.25)$$

3.2.4 Inverse Kinematics

Berikut ini penjelasan Permasalahan inverse kinematics tentang bagaimana kecepatan yang diberikan pada roda dan konfigurasi posisi awal robot $(x, y, \theta)_{t=0}$ dapat menentukan posisi robot (x, y, θ) . Diasumsikan $V_{right}(t) = V_{right}$, $V_{left}(t) = V_{left}$ dan $(x, y, \theta)_{t=0} = (0, 0, 0)$. Seperti Persamaan 3.26.

$$x(t) = \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} \sin \left[\frac{t}{d} (V_{right} - V_{left}) \right]$$

$$y(t) = -\frac{d(V_{right}+V_{left})}{2(V_{right}-V_{left})} \cos \left[\frac{t}{d} (V_{right} - V_{left}) \right] + \frac{d(V_{right}+V_{left})}{2(V_{right}-V_{left})}$$

$$\theta(t) = \frac{d(V_{right}+V_{left})}{2(V_{right}-V_{left})} \quad (3.26)$$

Diberikan waktu tujuan (t) dan posisi target (x,y), pada Persamaan 3.26 menyederhanakan V_{right} dan V_{left} , tetapi tidak menyediakan kontrol sendiri pada θ .

Persamaan 3.24 dan Persamaan 3.25 melakukan strategi sederhana untuk mendorong robot ke posisi tujuan untuk mendorong robot untuk beberapa tujuan menimbulkan (x, y, θ), robot bisa berputar sampai ke tempat tujuan pada(x, y), kemudian berjalan foward sampai titik (x, y), dan kemudian memutar di tempat sampai orientasi tujuan yang diperlukan θ terpenuhi.

3.3 Perancangan *Path Planning* dengan Metode *Radial Basis Function*

Perencanaan lintasan atau path planning dalam penelitian ini digunakan metode radial basis function. Radial basis function merupakan turunan dari struktur radial basis function network yang merupakan sebuah metode yang mempunyai struktur neural network yang terdiri dari 3 layer, yang meliputi :

- Input nodes layer
- Hidden neuron layer.yang berguna sebagai transformasi non-linier dari input melalui RBF
- Output layer, yang merupakan kombinasi linier dari output hidden layer

Persamaan dari Radial Basis Function untuk interpolasi adalah sebagai berikut seperti dijabarkan pada persamaan 3.27:

$$y = C(i) e^{\left(\frac{x-xa(k)}{\sigma_j} \right)^2} \quad (3.27)$$

Dimana :

y = output titik y yang dihasilkan, disebut sebagai radial basis. Persamaan (3.27) adalah fungsi Gaussian.

C = adalah titik center / pusat dari fungsi Gaussian

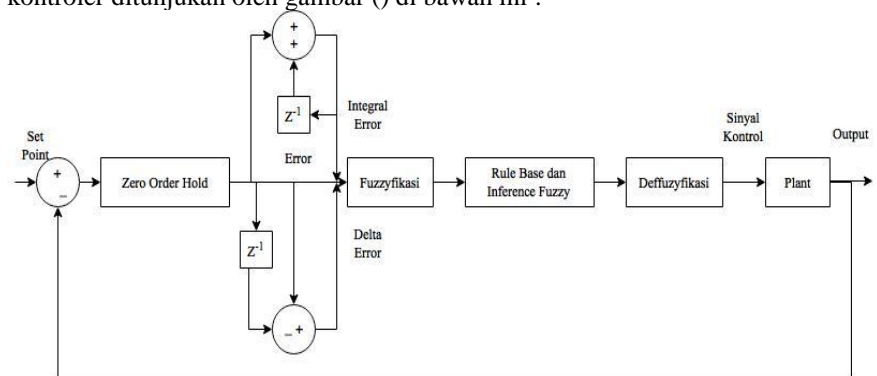
x = adalah input titik x.

σ = adalah lebar fungsi Gaussian hingga titik C.
 $i,j,k = 1,2,3,\dots,N$

3.4 Perancangan Kontroler *PID-Fuzzy*

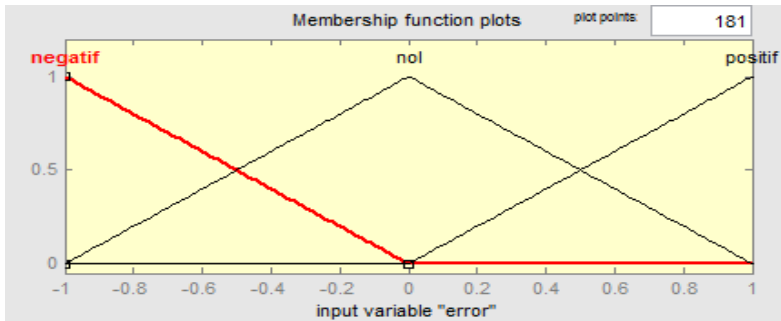
Prosedur melakukan desain kontroler PID-Fuzzy berupa menetapkan himpunan pendukung (baik input maupun output), melakukan operasi perhitungan fuzzyfikasi, mendefinisikan rule base dan inference fuzzy, serta melakukan operasi perhitungan defuzzyfikasi.

Nilai membership function dari kontroler PID-Fuzzy diperoleh dari data sensor yang sudah diuji pada respon kecepatan dari mobile robot. Membership function inilah yang nantinya akan menjadi parameter penentuan konstanta pada kontroler PID. Secara umum, diagram blok kontroler ditunjukkan oleh gambar () di bawah ini :



Gambar 3.4 Diagram Blok Kontroler *PID – Fuzzy*

Pada gambar 3.4 masukan dari kontroler PID Fuzzy adalah error, integral error, dan delta error seperti pada gambar 3.5 himpunan membership function fuzzy sebagai berikut:



Gambar 3.5 Himpunan Keanggotaan masukan *error*

Pada gambar 3.5 adalah himpunan keanggotaan error, sedangkan untuk delta error serta integral error kurang lebih sama. Membership function kemudian dibentuk aturan fuzzy seperti table (3.1) sampai table (3.3) di bawah ini:

Tabel 3.1 Rule Base dengan Delta Error Negatif

Delta error = Negatif

error	Integral error		
	1	2	3
	2	3	3
	3	3	3

Tabel 3.2 Rule Base dengan Delta Error Nol

Delta error = Nol

error	Integral error		
	1	1	2
	1	2	3
	2	3	3

Tabel 3.3 Rule Base dengan Delta Error Positif

Delta error = Positif

error	Integral error		
	1	1	1
	1	1	2
	1	2	3

Dimana :

1 = sinyal kontrol lambat

2 = sinyal kontrol sedang

3 = sinyal kontrol cepat

Nilai – nilai rule base merupakan nilai keluaran yang akan menjadi nilai konstanta sinyal kontrol setelah melalui operasi perhitungan inference fuzzy serta defuzzifikasi.

3.5 Perancangan Software

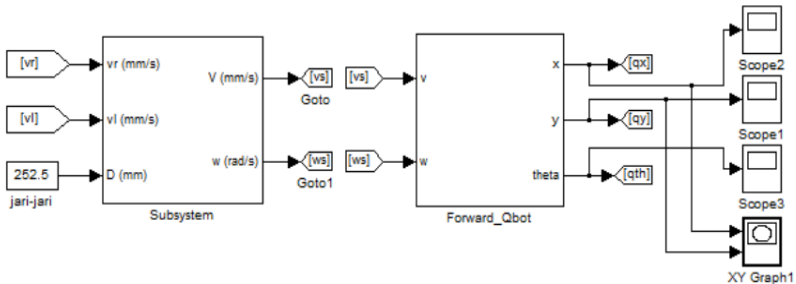
Perancangan perangkat lunak untuk perencanaan lintasan pada Qbot menggunakan Matlab. Perangkat lunak terbagi menjadi dua bagian yaitu simulasi dan implementasi. Pada simulasi, perancangannya meliputi perencanaan lintasan pada Qbot dengan menggunakan *Radial Basis Function* dan perancangan kontroler *PID Fuzzy*. Perangkat lunak implementasi dirancang untuk membuktikan dan pengujian perencanaan lintasan pada Qbot dengan *Radial Basis Function* dan sebagai kontrolernya yaitu *PID Fuzzy*.

3.5.1 Perancangan Software Simulasi

Pada perancangan perangkat lunak simulasi yang akan dirancang terdiri dari pemodelan plant, perencanaan lintasan dengan *Radial Basis Function*, dan merancang kontroler *PID Fuzzy*.

3.5.1.1 Pemodelan Plant dengan Simulasi

Pemodelan Plant dengan simulasi diperoleh dari persamaan sistem fisik yang dibahas pada sub bab sebelumnya yaitu pada 3.1.

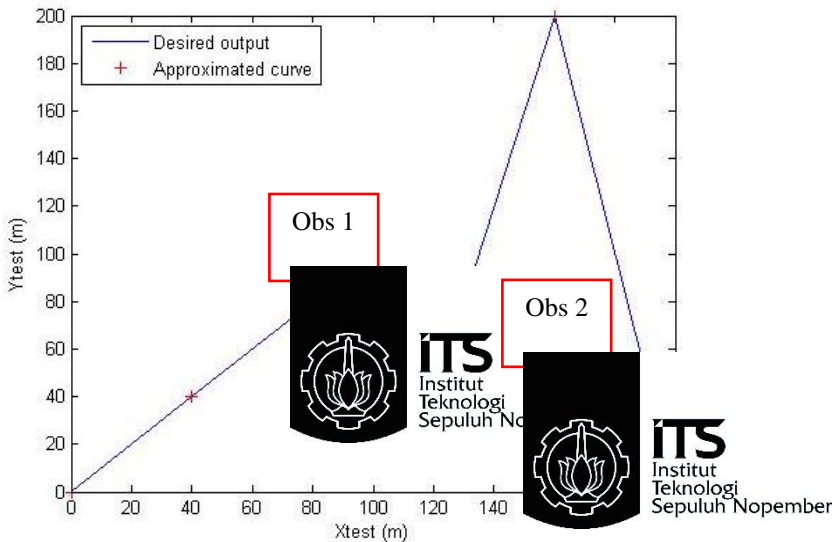


Gambar 3.6 Blok *Simulink* Simulasi *Mobile Robot Qbot*

Untuk blok plant terdiri dari kecepatan roda kanan dan kiri serta diameter dari Qbot yang nantinya akan menghasilkan keluaran berupa jumlah kecepatan roda kanan ($V_{mm/s}$) dan kecepatan roda kiri ($V_{mm/s}$) serta kecepatan sudut ($\omega \text{ rad/s}$), V dan ω akan menentukan nilai x , y dan θ Qbot.

3.5.1.2 Perencanaan Lintasan dengan *Radial Basis Function* Simulasi

Pada perencanaan simulasi pada lintasan robot menggunakan *Radial Basis Function* dengan code pada Matlab untuk merencanakan lintasannya dengan menentukan grid sesuai yang kita inginkan, menentukan obstacles dan lalu mendapatkan hasil lintasan yang akan digunakan pada Qbot. Output pada persamaan radial basis yaitu titik titik Y akan dicari dari pendekatan radial basis, sedangkan input adalah titik titik X . Kemudian menentukan lebar pita dari fungsi radial basis yang sudah ditentukan. Pada Gambar 3.7 adalah simulasi hasil lintasan dengan *Radial Basis Function*.



Gambar 3.7 Simulasi Perencanaan Lintasan *Radial Basis Function*

Pada perencanaan lintasan di atas ditentukan grid yaitu 0x200 dan tentukan jumlah obstacles yaitu 2 buah. Sehingga dapat diketahui lintasan pada *Radial Basis Function*. Dengan memasukkan parameter input yang berupa titik titik perkiraan agar menghindari obstacles berupa titik x dan y. Akan didapatkan pendekatan dari fungsi *radial basis function*.

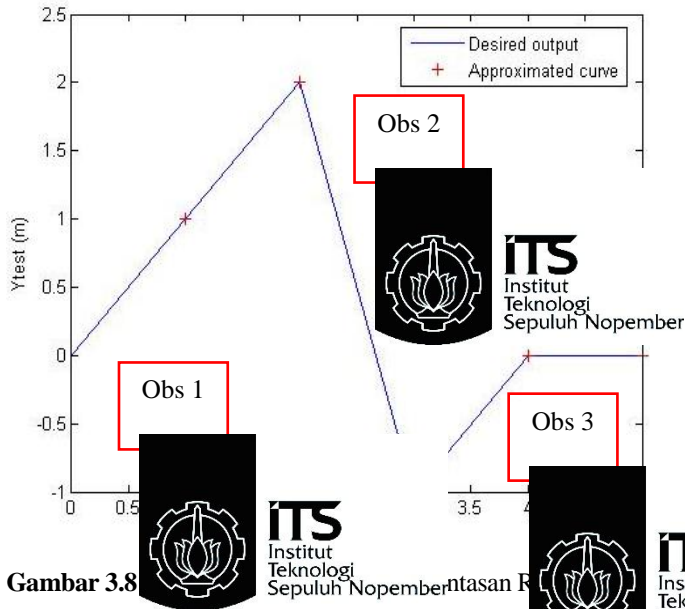
TUGAS AKHIR - TE 141599

3.5.2 Perancangan Software Implementasi Perencanaan Jalur dan

Pada perancangan perancangan jalur robot menggunakan metode *Radial Basis Function* dengan perangkat lunak Matlab Simulink yang terpasang perangkat lunak QuARC. Dengan perangkat lunak QuARC, Qbot dapat terhubung dengan Matlab Simulink sehingga program yang telah dirancang dapat diupload langsung di unduh pada Qbot. Selain itu, parameter yang telah dirancang dapat langsung dimasukkan pada Qbot secara *real time* dan data-data yang akan dirancang terdiri dari pembuatan lintasan robot dengan *Radial Basis Function* (secara *offline*) untuk mengeksekusi robot dari posisi

Dosen Pembimbing Welly Julian
Ir. Rusdhianto Effendi A.K., M.P.

awal sampai ke target yang di inginkan sesuai dengan simulasi yang sudah dibuat. Sehingga perencanaan lintasan yang kita buat di sesuaikan.



Gambar 3.8

Pada implementasi perencanaan lintasan yang di buat, posisi awal Qbot mobile robot kita tentukan sesuai dengan simulasi. Dimana berupa titik-titik waypoint yang telah di dapat dan obstacles pada lintasan robot. Misalkan direpresentasikan jarak dari posisi awal robot sampai target robot berjarak 5m dan melalui beberapa titik yang telah didapat pada simulasi..

Pada Gambar 3.9 dimana Qbot mobile robot menggunakan kecepatan roda sebagai input dan output data baru robot serta memberikan memberikan data berupa data referensi robot dengan data real robot setelah dilakukan pengujian .

Blok *path planner* berfungsi untuk menghasilkan perintah gerak yang tepat untuk mengikuti lintasan yang kita tentukan dan akan di eksekusi oleh Qbot. Dalam metode perencanaan lintasan robot dari posisi awal menuju posisi target sesuai dengan lintasan yang ditentukan

TUGAS AKHIR - TE 141599
IMPLEMENTASI PERENCANAAN JALUR DAN KENDALI
MOBILE ROBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROL

IMPLEMENTASI PERENCANAAN JALUR DAN KENDALI
MOBILE ROBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROL

IMPLEMENTASI PERENCANAAN JALUR DAN KENDALI
MOBILE ROBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROL

IMPLEMENTASI PERENCANAAN JALUR DAN KENDALI
MOBILE ROBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROL

IMPLEMENTASI PERENCANAAN JALUR DAN KENDALI
MOBILE ROBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROL

IMPLEMENTASI PERENCANAAN JALUR DAN KENDALI
MOBILE ROBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROL

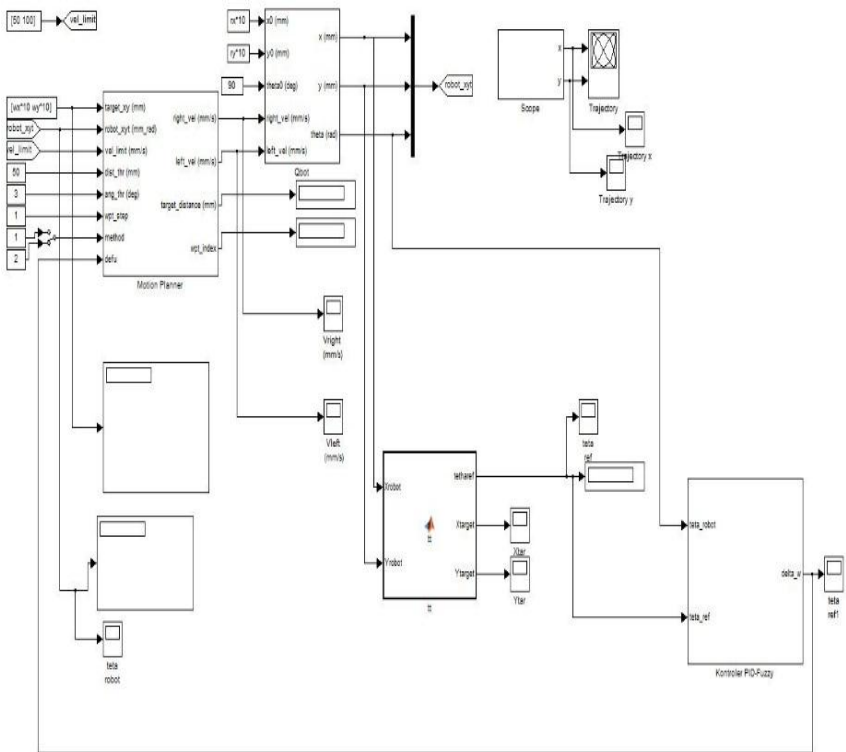
IMPLEMENTASI PERENCANAAN JALUR DAN KENDALI
MOBILE ROBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROL

Welly Julian
 NRP. 2214 105 024

35
 JURUSAN TEKNIK ELEKTRO
 Fakultas Teknologi Industri
 Institut Teknologi Sepuluh Nopember
 Surabaya 2019 Welly Julian

yaitu berupa titik waypoint pada simulasi *Radial Basis Function* yang di dapat.

Kontroler yang dirancang pada implementasi untuk Qbot ini menggunakan metode kontroler logika Fuzzy tipe PID. Kontroler ini digunakan untuk mengatur kecepatan gerak roda kanan, roda kiri, dan sudut robot Qbot.



Gambar 3.9 Blok *Simulink* Implementasi Qbot

BAB 4

HASIL DAN ANALISA

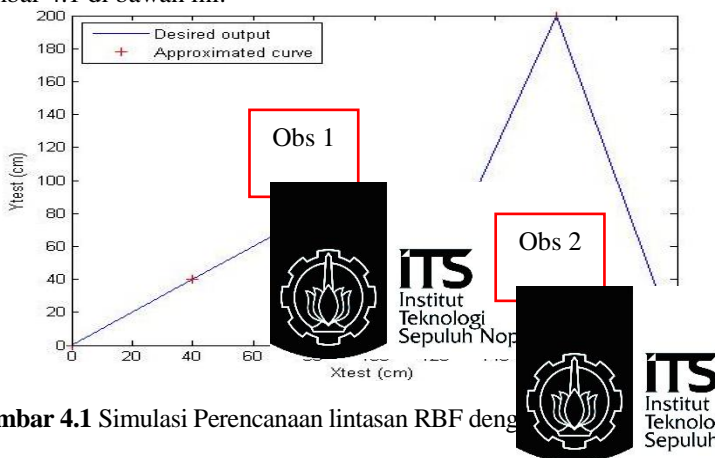
Pada bab ini membahas mengenai implementasi metode *Radial Basis Function* dengan sistem kontrol *PID-fuzzy* yang telah dirancang pada bab sebelumnya. Untuk implementasi kontroler *PID-fuzzy* terlebih dahulu dilakukan pengambilan data training dengan menggunakan metode *radial basis function* sebagai perancang lintasan atau lintasan untuk output yang diinginkan pada plant. Selanjutnya *output* tersebut akan diimplementasikan untuk pergerakan plant yang diberi kontroler *PID-fuzzy*.

4.1 Simulasi Perencanaan Lintasan dengan *Radial Basis Function*

Pada tahap simulasi akan ditampilkan rancangan lintasan yang akan dilalui robot, serta data *trajectory* yang didapat dari ω_x dan ω_y , dan ilustrasi dari *training* mobile robot.

4.1.1 Simulasi Perencanaan lintasan RBF dengan 2 obstacles

Pada tahap simulasi perencanaan lintasan RBF dengan 2 obstacles dengan *radial basis function* ini ditentukan posisi awal serta posisi akhir robot dari koordinat x dan y. Juga jarak tepuh yang akan dilalui robot. Pada simulasi ini diasumsikan *grid map* 200 cm dari 0 hingga 200. Dimana posisi awal robot pada $(x,y) = (0,0)$ dan posisi target $(x,y) = (200,0)$. Untuk simulasi dari *mobile robot* dapat dilihat pada gambar 4.1 di bawah ini.



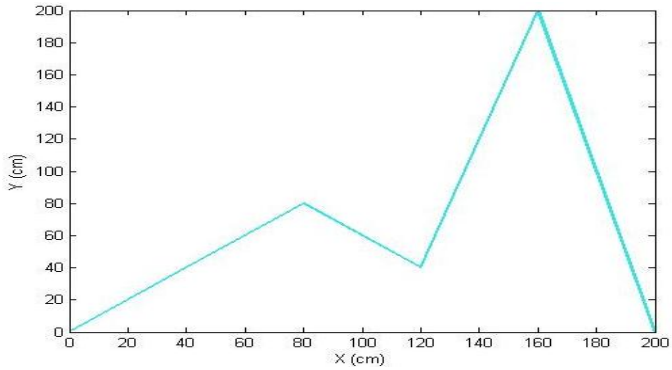
Gambar 4.1 Simulasi Perencanaan lintasan RBF dengan 2 obstacles

Ilustrasi dari *training mobile robot* dapat dilihat pada gambar 4.2.



Gambar 4.2 Posisi Robot Training Lintasan 1

Simulasi *trajectory* dari hasil dari ω_x dan ω_y dapat dilihat pada gambar 4.3

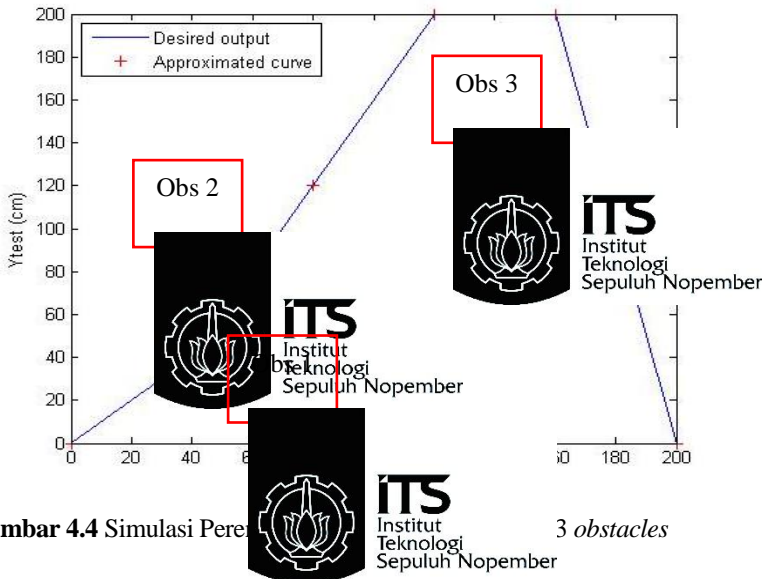


Gambar 4.3 Hasil *Trajectory* Perencanaan lintasan RBF dengan 2 *obstacles*

Pada gambar 4.3 didapatkan data dari ωx dan ωy untuk *input* dari Map 1 untuk *Simulink* yang digunakan untuk simulasi lintasan 1.

4.1.2 Simulasi Perencanaan lintasan RBF dengan 3 obstacles

Pada tahap simulasi perencanaan lintasan RBF dengan 3 obstacles dengan *radial basis function* ini ditentukan posisi awal serta posisi akhir robot dari koordinat x dan y. Juga jarak tepuh yang akan dilalui robot. Pada simulasi ini diasumsikan *grid map* 200 cm dari 0 hingga 200. Dimana posisi awal robot pada $(x,y) = (0,0)$ dan posisi target $(x,y) = (200,0)$. Untuk simulasi dari *mobile robot* dapat dilihat pada gambar 4.4 di bawah ini.



Gambar 4.4 Simulasi Perencanaan lintasan RBF dengan 3 obstacles

Ilustrasi dari *training mobile robot* dapat dilihat pada gambar 4.5.

TUGAS AKHIR - TE 141599

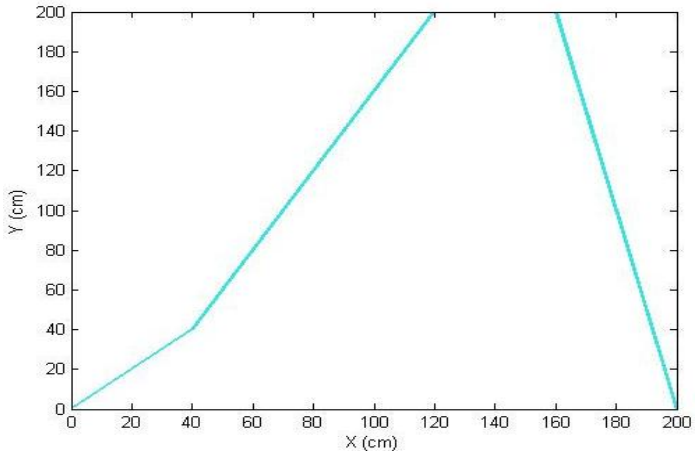
IMPLEMENTASI PERENCANAAN
TUGAS AKHIR - TE 141599 MOBILE ROBOT QBOT MENGGUNAKAN
ALGORITMA GENETIKA DAN KONTROLER FUZZY
IMPLEMENTASI PERENCANAAN JALUR DAN KENDAL
MOBILE ROBOT QBOT MENGGUNAKAN METODE
ALGORITMA GENETIKA DAN KONTROLER FUZZY

TUGAS AKHIR - TE 141599 Welly Julian



Gambar 4.5 Posisi Robot *Training* Lintasan 2

Simulasi *trajectory* dari hasil dari ω_x dan ω_y dapat dilihat pada gambar 4.6.



Gambar 4.6 Hasil *Trajectory* Perencanaan lintasan RBF dengan 3 *obstacles*

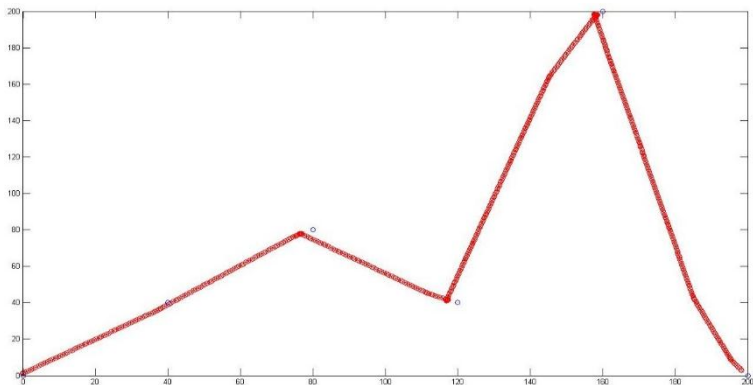
Pada gambar 4.6 didapatkan data dari ω_x dan ω_y untuk input dari Map 1 untuk Simulink yang digunakan untuk simulasi perencanaan lintasan RBF dengan 3 *obstacles*.

4.2 Implementasi Trayektori dengan Kontroler PID Fuzzy

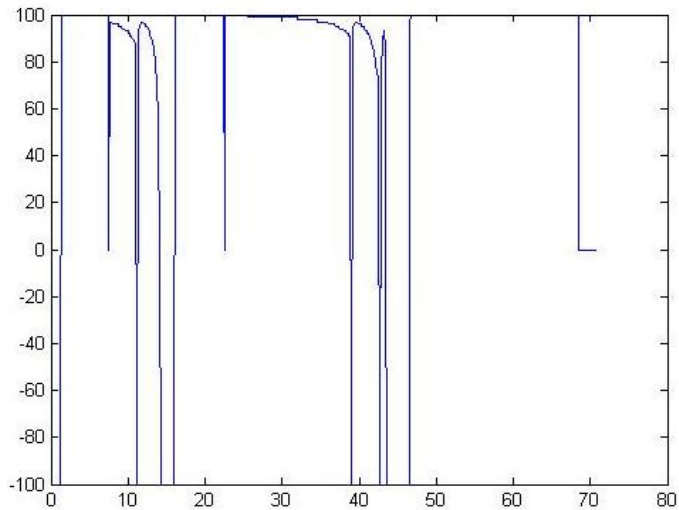
Implementasi perencanaan lintasan RBF dengan 2 obstacles dan perencanaan lintasan RBF dengan 3 obstacles menggunakan *radial basis function* ini dilakukan sebanyak 2 kali percobaan. Percobaan dilakukan sebanyak 2 kali untuk membandingkan hasil dari kontrol ini dapat bekerja dengan baik atau tidak. Dari setiap percobaan data yang diambil untuk dibandingkan yaitu kecepatan roda kanan, kecepatan roda kiri serta kontrol *PID Fuzzy*.

4.2.1 Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles

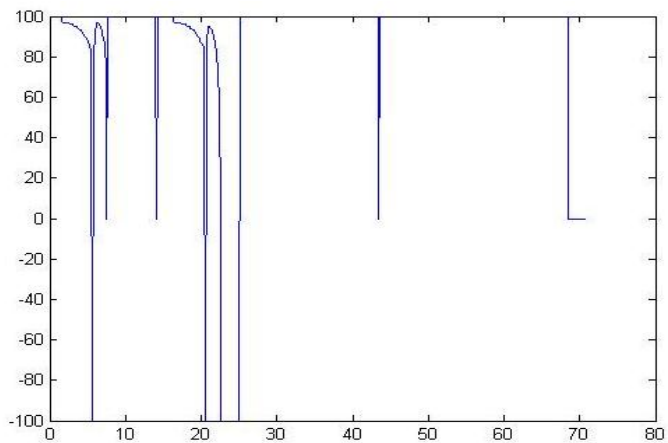
Hasil dari implementasi dari percobaan implementasi trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles dengan set point kecepatan 100 untuk *mobile robot*, terdiri dari hasil *track*, kecepatan roda kanan, kecepatan roda kiri, ω_x , dan ω_y dapat dilihat pada gambar 4.7, 4.8, 4.9, 4.10, dan 4.11 dibawah ini.



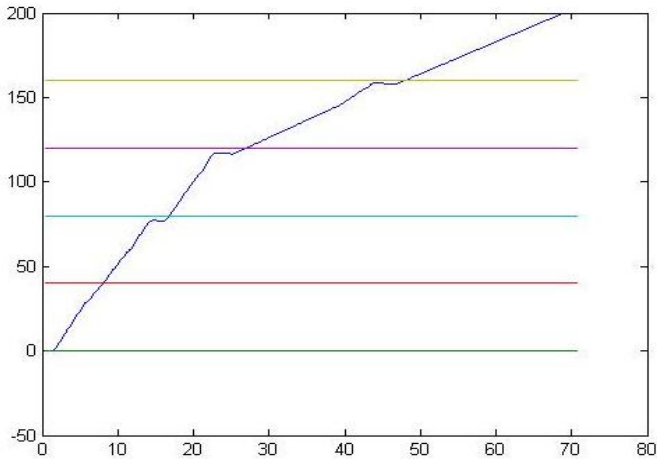
Gambar 4.7 Hasil *Track* Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles



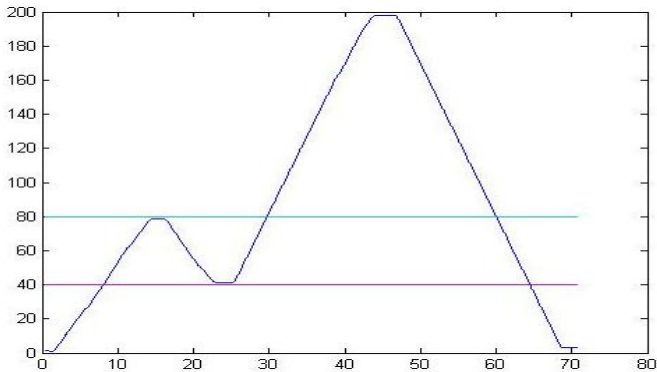
Gambar 4.8 Kecepatan Roda Kanan Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles



Gambar 4.9 Kecepatan Roda Kiri Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles



Gambar 4.10 Respons ω_x Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles



Gambar 4.11 Respons ω_y Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles

Pada implementasi trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles data yang didapat adalah waktu yang dipakai robot mulai dari titik awal hingga titik tujuan adalah 68,70 detik. Didapatkan dari data dari workspace Matlab baik dari kecepatan roda kanan, kecepatan roda kiri, ω_x , dan ω_y .

Pada table 4.1 , 4.2, 4.3, dan 4.4 berikut akan ditampilkan beberapa data dari kecepatan roda kanan, kecepatan roda kiri, ω_x , dan ω_y .

Tabel 4.1 Kecepatan Roda Kanan

Waktu (detik)	Kecepatan (konstan)
0.300	-100
0.400	-100
0.500	-100
0.700	-100

Tabel 4.2 Kecepatan Roda Kiri

Waktu (detik)	Kecepatan (konstan)
0.300	100
0.400	100
0.500	100
0.700	100

Tabel 4.3 Kecepatan pada sumbu x

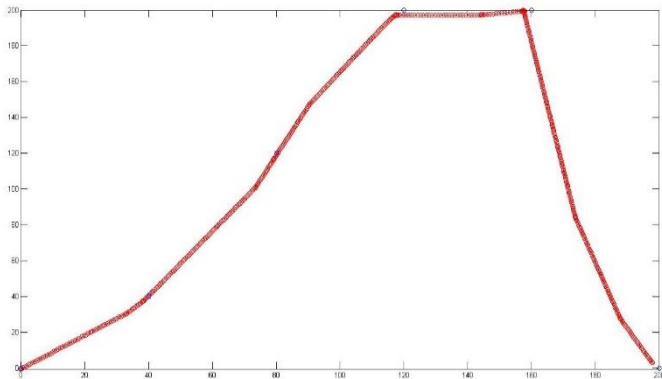
Waktu (detik)	Kecepatan (konstan)
0.300	-0,0017
0.400	-0,0017
0.500	-0,0156
0.700	-0,0156

Tabel 4.4 Kecepatan pada sumbu y

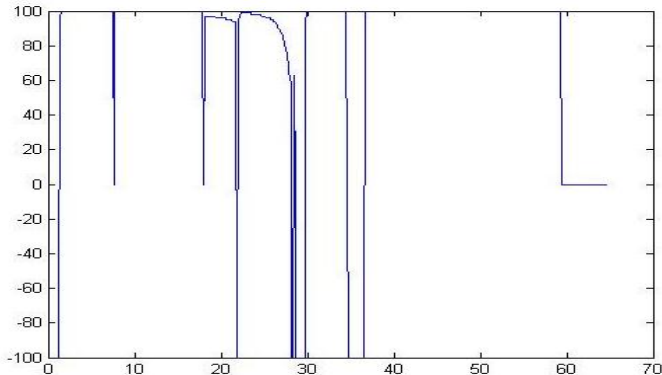
Waktu (detik)	Kecepatan (konstan)
0.300	1,4000
0.400	1,4000
0.500	1,3009
0.700	1,3009

4.2.2 Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles

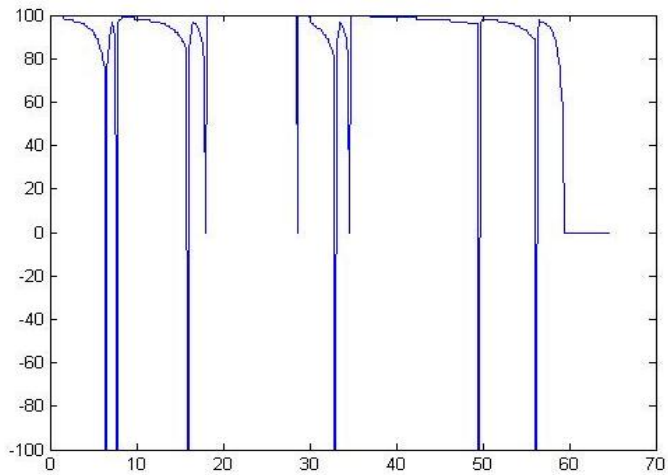
Hasil dari implementasi dari percobaan implementasi trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles dengan set point kecepatan 100 untuk *mobile robot* , terdiri dari hasil track lintasan 2, kecepatan roda kanan, kecepatan roda kiri, ω_x , dan ω_y dapat dilihat pada gambar 4.12, 4.13, 4.14, 4.15, dan 4.16 dibawah ini.



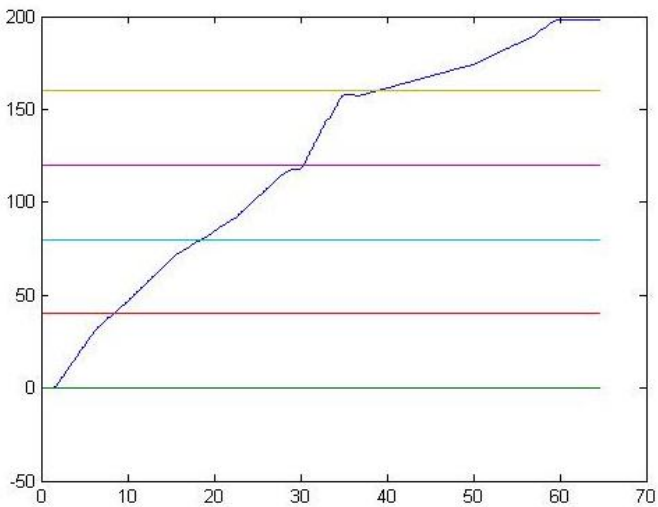
Gambar 4.12 Hasil Track Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles



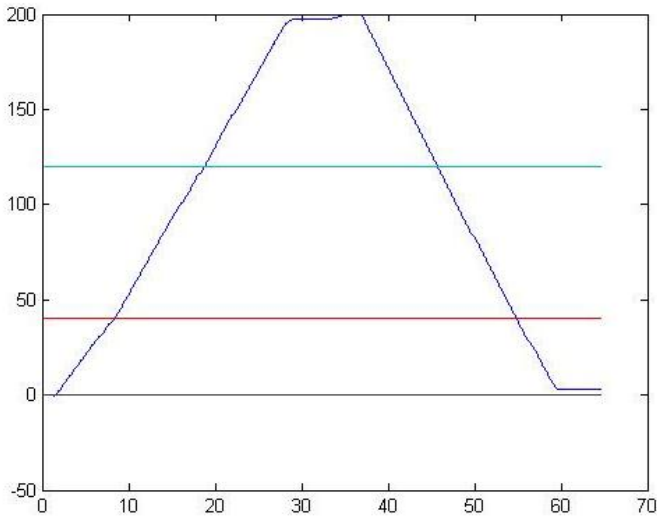
Gambar 4.13 Kecepatan Roda Kanan Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles



Gambar 4.14 Kecepatan Roda Kiri Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles



Gambar 4.15 Respons ω_x Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles



Gambar 4.16 Respons ω_y Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles

Pada implementasi trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles data yang didapat adalah waktu yang dipakai robot mulai dari titik awal hingga titik tujuan adalah 59,40 detik. Didapatkan dari data dari *workspace Matlab* baik dari kecepatan roda kanan, kecepatan roda kiri, ω_x , dan ω_y .

Pada table 4.5 , 4.6, 4.7, dan 4.8 berikut akan ditampilkan beberapa data dari kecepatan roda kanan, kecepatan roda kiri, ω_x , dan ω_y .

Tabel 4.5 Kecepatan Roda Kanan

Waktu (detik)	Kecepatan (konstan)
0.300	-100
0.400	-100
0.500	-100
0.700	-100

Tabel 4.6 Kecepatan Roda Kiri

Waktu (detik)	Kecepatan (konstan)
0.300	100
0.400	100
0.500	100
0.700	100

Tabel 4.7 Kecepatan pada sumbu x

Waktu (detik)	Kecepatan (konstan)
0.300	0
0.400	-0,0034
0.500	-0,0034
0.700	-0,0034

Tabel 4.8 Kecepatan pada sumbu y

Waktu (detik)	Kecepatan (konstan)
0.300	0
0.400	-0,0999
0.500	-0,0999
0.700	-0,0999

4.3 Implementasi Kontroler *PID Fuzzy*

Pada implementasi kontroler *PID Fuzzy* didapatkan *input* berupa error, delta error, dan integral error. Kemudian output dari kontroler ini berupa sinyal control $u(k)$.

Pada table 4.9 dan 4.10 dibawah ini akan ditunjukkan hasil dari kontroler *PID Fuzzy* untuk lintasan 1 dan lintasan 2.

Tabel 4.9 Hasil Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 2 Obstacles

No.	Waktu (det)	Error	Delta error	Integral Error	Sinyal Kontrol
1.	0.4	-0,7857	0,0187	-3,9846	0,0000
2.	0.5	-0,7334	0,0523	-4,7181	0,0000
3.	0.6	-0,6624	0,0709	-5,3806	0,0000
4.	0.7	-0,6101	0,0523	-5,9907	0,0000
5.	0.8	-0,5228	0,0872	-6,5136	0,00068
6.	0.9	-0,4522	0,0705	-6,9658	0,00127

Tabel 4.10 Hasil Implementasi Trayektori dengan Kontroler PID Fuzzy dengan 3 Obstacles

No.	Waktu (det)	Error	Delta error	Integral Error	Sinyal Kontrol
1.	0.4	-0,7853	0	-3,1415	0,0000
2.	0.5	-0,7492	0,0361	-3,8908	0,0000
3.	0.6	-0,7143	0,0349	-4,6052	0,0000
4.	0.7	-0,6445	0,0698	-5,2498	0,0000
5.	0.8	-0,5737	0,0707	-5,8236	0,00063
6.	0.9	-0,5039	0,0698	-6,3276	0,00127

Keluaran dari sinyal kontrol *PID Fuzzy* akan dikalikan dengan *gain* sehingga menjadi *delta omega* ($\Delta\omega$). Maka untuk menjadi masukan dari *plant* Qbot dengan *Set point/Vlimit* (min 50 max 100) untuk *vr* dan *vl* menjadi :

$$V_r = V_{limit} - \Delta\omega$$

$$V_l = V_{limit} + \Delta\omega.$$

(halaman ini sengaja dikosongkan)

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil pengimplementasian perencanaan lintasan pada mobile robot Qbot menggunakan metode Radial Basis Function dengan kontroler PID Fuzzy dapat disimpulkan bahwa plant mobile robot dapat mengikuti data referensi dari hasil keluaran metode RBF dengan output titik x dan y yang telah dilakukan. Sebagaimana dari hasil pengujian perencanaan lintasan dengan metode RBF dan kontroler PID fuzzy pada Qbot mobile robot untuk implementasi trayektori dengan kontroler PID Fuzzy dengan 2 Obstacles dengan set point sebesar 100, waktu yang dibutuhkan untuk mencapai target adalah 68,70 detik untuk bergerak dari titik (0,0) cm menuju titik (0,200) cm. Untuk implementasi trayektori dengan kontroler PID Fuzzy dengan 3 Obstacles saat implementasi membutuhkan waktu 59,40 detik untuk bergerak dari titik (0,0) cm menuju titik (0,200) cm. Untuk perbedaan respon sinyal kontrol PID fuzzy tergantung pada input error, delta error, dan integral error. Untuk implementasi trayektori dengan kontroler PID Fuzzy dengan 2 Obstacles nilai error rata rata adalah 0,0887 yang hanya sedikit berpengaruh pada tetha robot untuk menentukan kecepatan roda kiri serta roda kanan. Sedangkan implementasi trayektori dengan kontroler PID Fuzzy dengan 3 obstacles nilai error rata rata adalah 0,0235 yang hanya sedikit berpengaruh pada tetha robot untuk menentukan kecepatan roda kiri serta roda kanan..

5.2 Saran

Untuk pengembangan penelitian mengenai *Qbot mobile robot* diharapkan pada penelitian selanjutnya dapat mengembangkan pergerakan *Qbot mobile robot* ini menggunakan kontroler lain yang sifatnya adaptive dengan dilengkapi komponen lain yang mendukung kinerja robot seperti menggunakan kamera *optick track* untuk proses trayektori.

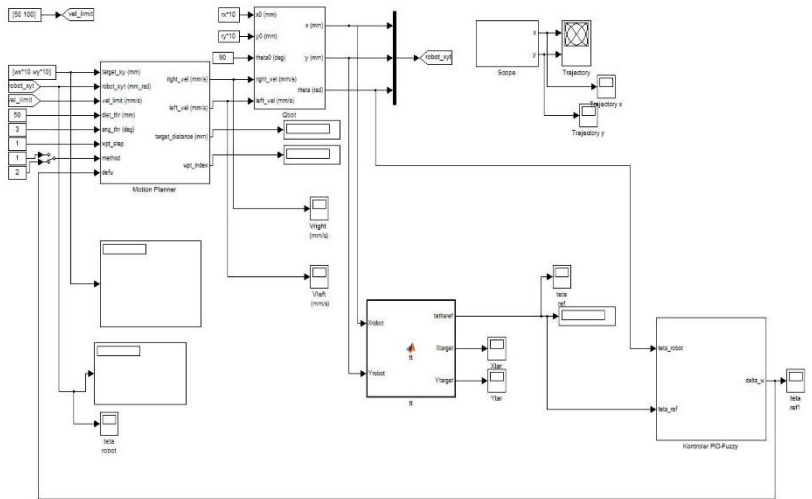
(halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

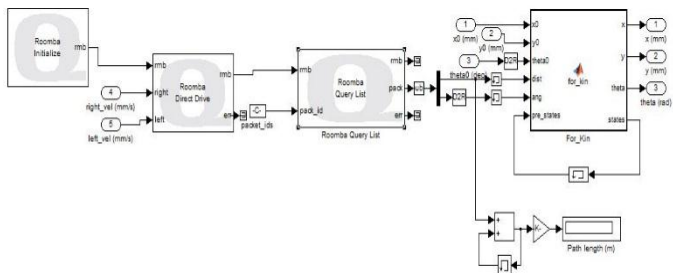
- [1] Dudek Gregory, Jenkin Michael, "*Computational Principles of Mobile Robotics*", 2005.
- [2] Mlynek D.M, Patyra M.J "*Fuzzy logic implementation and applications*" Wiley, Teubner, 1996.
- [3] Molionis, Georgios, "*A Framework for Collaborative Quadrotor - Ground Robot Mission*", Naval Postgraduate School, Monterey, California ,2011.
- [4] D'Amico A, Ippoliti. G, S. Longhi, "*A Radial Basis Function Network Approach for the Tracking Problem of Mobile Robot*", Dipartimento d Electronics, Bianchi, Italy, 2008.
- [5] Stentz, Anthony, "*Optimal efficient Path Planning for partially Known Environment*", Robotic Institute, Pittsburgh, Russia, 2007
- [6] _____, "*User Manual Quanser Qbot*", Inovative Educate, Quanser, Document number 830, Revision 7, 2012
- [7] _____, "*Kinematics Modeling*", Inovative Educate, Quanser, Document number 830, Revision 7, 2012.
- [8] Mark, W. Spong., "*Robot Dynamic and Control*", John Willey and Sons, 2005
- [9] Chen, Jiajia. Zhao, Pan. Liang, Huawei. Mei, Tao, "*Motion Planning for Autonomous Vehicle Based on Radial Basis Function Neural Network in Unstructured Environment*", School of Engineering Science, University of Science and Technology of China, Hefei 230026, China, 2014

(halaman ini sengaja dikosongkan)

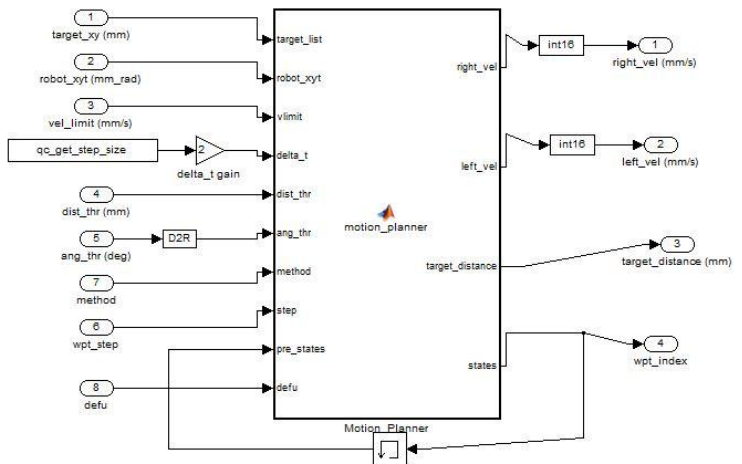
1. LAMPIRAN A



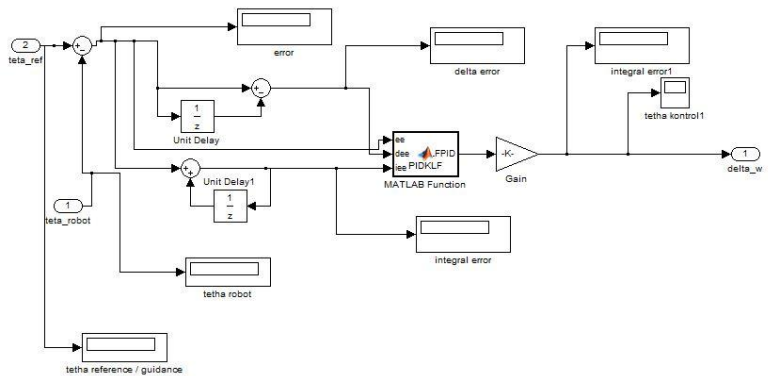
Gambar A1. Blok Keseluruhan Program



Gambar A2. Blok Model Qbot



Gambar A3. Blok Motion Planner



Gambar A4. Blok Kontroler *PID Fuzzy*

2. LAMPIRAN B

1. Program Kinematik Robot

```
function [right_vel, left_vel, target_distance, states] = ...
    motion_planner(target_list, robot_xyt, vlimit, ...
        delta_t, dist_thr, ang_thr, method, step, pre_states)

% Initialize output variables
right_vel = int16(0);
left_vel = int16(0);
states = pre_states;

rx = robot_xyt(1);
ry = robot_xyt(2);
rtheta = check_angle(robot_xyt(3));
target_xy = [rx ry];
[n xy] = size(target_list);
if n==2 && xy==1
    target_xy = [target_list(1,1) target_list(2,1)];
    n = 1;
else
    for i=1:n
        if states(1) == i
            for j=1:xy
                target_xy(j) = target_list(i, j);
            end
        end
    end
end

tx = target_xy(1);
ty = target_xy(2);
target_distance = find_dist(rx, ry, tx, ty);

if ((states(1) == n) && (target_distance <= dist_thr)) %|| states(1) ==
-1
%     states(1) = -1;
    right_vel = int16(0);
```

```

        left_vel = int16(0);
%        target_distance = 0;
else
    if target_distance <= dist_thr
        states(1) = pre_states(1) + step;%1;
        if states(1) > n
            states(1) = n;
        end
    else
        [ang_to_tar]= find_theta(rx, ry, rtheta, tx, ty);
        if method == 2
            [right_vel, left_vel] = solve_inv_kin(target_distance, ...
                ang_to_tar, vlimit, delta_t);
        else
            [right_vel, left_vel] = rotate_and_go(ang_to_tar, ...
                vlimit, ang_thr);
        end
    end
end
end
end

return;

%-----
function [y] = check_angle(x)
y = x;
if x > pi
    y = x - 2*pi;
elseif x < -pi
    y = x + 2*pi;
end
return;
% -----
function dist = find_dist(rx, ry, tx, ty)
dist = sqrt((rx-tx)^2 + (ry-ty)^2);
return;
% -----
function [theta]= find_theta(rx, ry, rtheta, tx, ty)

```



```

X = tx - rx;
Y = ty - ry;
theta = atan2(Y, X);
theta = check_angle(theta - check_angle(rtheta));

return;
% -----
function [vr, vl] = solve_inv_kin(dist, theta, vlimit, delta_t)

d = 252.5;
vmax = vlimit(2);
wmax = (2*vmax)/d;
w = theta/delta_t;
w_sign = sign(w);
if abs(w) > wmax
    w = w_sign*wmax;
    vr = int16(round((d*w)/2));
    vl = int16(-vr);
else
    v = dist/delta_t;
    vr_tmp = (2*v + d*w)/2;
    vl_tmp = 2*v - vr_tmp;

    max_of_vrvl = abs(max(vr_tmp, vl_tmp));
    if max_of_vrvl > vmax
        vr_tmp = (vr_tmp/max_of_vrvl)*vmax;
        vl_tmp = (vl_tmp/max_of_vrvl)*vmax;
    end
    vr = int16(vr_tmp);
    vl = int16(vl_tmp);

end

return;
% -----
function [vr, vl] = rotate_and_go(theta, vlimit, ang_thr)

if abs(theta) > ang_thr

```

```

    if theta >= 0
        vr = int16(vlimit(1));
        vl = int16(-vlimit(1));
    else
        vr = int16(-vlimit(1));
        vl = int16(vlimit(1));
    end
else
    vr = int16(vlimit(2));
    vl = int16(vlimit(2));
end

return;
% -----

```

2. Program Motion Robot

```

function [x, y, theta, states] = for_kin(x0, y0, theta0, dist, ang,
pre_states)
% This block supports an embeddable subset of the MATLAB
language.
% See the help menu for details.

states = pre_states;

if states(1) == 0
    x = x0;
    y = y0;
    theta = check_angle(theta0);
    states(1) = 1;
else
    theta = check_angle(check_angle(states(4)) + check_angle(ang));
    x = states(2) + dist*cos(theta);
    y = states(3) + dist*sin(theta);
end

states(2) = x;
states(3) = y;
states(4) = theta;

```

```

return;

function y = check_angle(x)
y = x;
if y > pi
    y = x - 2*pi;
elseif y < -pi
    y = x + 2*pi;
end
return;

```

3. Program Mendapatkan Trajectory

```

clear all
close all
clc

% Initialize variables -----
load MAP1
method = 'Radial Basis';
X = 1:mapwidth;
X = X - maporgx;
Y = 1:mapheight;
Y = Y - maporgy;
maptitle = 'Occupancy map';
n = length(obs);

% Plot occupancy map -----
figure(4)
% plot(X, zeros(1, mapwidth)),
hold on
% plot(zeros(1, mapwidth), Y)
axis([min(X) max(X) min(Y) max(Y)])
grid minor
title(maptitle);

% Plot robot's position -----

```

```

rx = Rx - maporgx;
ry = -(Ry - maporgy);

% Plot target's position -----
tx = Tx - maporgx;
ty = -(Ty - maporgy);
rx=-200;
ry=-200;
tx=200;
ty=200;
% Find waypoints -----
[wx,wy]=rbfsatu(-200,-200,200,200);
wx=wx';
wy=wy';

% Plot robot's trajectory -----
figure(4)
plot([rx;wx], [ry;wy], 'LineWidth', 2, 'Color', [0.2 0.9 0.9])
xlabel('X (cm)')
ylabel('Y (cm)')
% text(rx+5, ry, 'Robot');
% text(tx+5, ty, 'Target');
grid off
hold off

```

4. Program RBF 1

```

function [Xx,Yy]=rbfduadua(a,b,c,d)
%Program Matlab untuk Radial Basis Function
clc;
clear all;
close all;
batas=10;
tengah=0.1;
% x=0:1:10;
% x1=10:1:20;
x=-200:80:200;
y=[-200 -120 -40 -120 200 -200];
%Menghasilkan data terlatih dengan nilai random

```

```

% y=2*sin(2*pi*0.1*x);
% y1=2*sin(pi*0.1*x1);
% for i=1:length(x)
%
%           y(i)=2*sin(2*pi*0.1*x);%2*sin(0.147*pi*x(i))-
1*sin(0.3125*pi*x(i))+2*sin(0.0000000001*pi*x(i));
% end

```

```

% Framing the interpolation matrix for training data
t=length(x);
for i=1:t
    for j=1:t
        h=x(i)-x(j);
        k=h^2/2.5;
        train(i,j)=exp(-k);
    end
end
W=inv(train)*y';

```

```

% Test RBF
xtest=-200:80:200;

```

```

% ytest output yang diinginkan
ytest=[-200      -120      -40      -120      200      -
200];%2*sin(2*pi*0.1*xtest);%2*sin(0.147*pi*xtest)-
1*sin(0.3125*pi*xtest)+2*sin(0.0000000001*pi*xtest);

```

```

% Framing the interpolation matrix for test data
t1=length(xtest);
t=length(x);
for i=1:t1
    for j=1:t
        h=xtest(i)-x(j);
        k=h^2/2.5;
        test(i,j)=exp(-k);
    end
end
actual_test=test*W;

```

```

% Plot hasil dari jaringan RBF

```

```

% plot(x,y,'b+-');
% plot(x,y,'b+-');
figure;plot(xtest,ytest,'b-',xtest,actual_test,'r+');
xlabel('Xtest (m)');
ylabel('Ytest (m)');
h = legend('Desired output','Approximated curve',2);
set(h);
Xx=xtest;
Yy=actual_test;

```

5. Program RBF 2

```

function [Xx,Yy]=rbfduadua(a,b,c,d)
%Program Matlab untuk Radial Basis Function
clc;
clear all;
close all;
batas=10;
tengah=0.1;
% x=0:1:10;
% x1=10:1:20;
x=-200:80:200;
y=[-200 -120 40 200 200 -200];
%Menghasilkan data terlatih dengan nilai random
% y=2*sin(2*pi*0.1*x);
% y1=2*sin(pi*0.1*x1);
% for i=1:length(x)
%
%           y(i)=2*sin(2*pi*0.1*x);% 2*sin(0.147*pi*x(i))-
% 1*sin(0.3125*pi*x(i))+2*sin(0.0000000001*pi*x(i));
% end

% Framing the interpolation matrix for training data
t=length(x);
for i=1:1:t
    for j=1:1:t
        h=x(i)-x(j);
        k=h^2/2.5;
        train(i,j)=exp(-k);
    end
end

```

```

end
end
W=inv(train)*y';

% Test RBF
xtest=-200:80:200;

% ytest output yang diinginkan
ytest=[-200      -120      40      200      200      -
200];%2*pi*(2*pi*0.1*xtest);%2*sin(0.147*pi*xtest)-
1*sin(0.3125*pi*xtest)+2*sin(0.0000000001*pi*xtest);

% Framing the interpolation matrix for test data
t1=length(xtest);
t=length(x);
for i=1:1:t1
    for j=1:1:t
        h=xtest(i)-x(j);
        k=h^2/2.5;
        test(i,j)=exp(-k);
    end
end
actual_test=test*W;

% Plot hasil dari jaringan RBF
% plot(x,y,'b+-');
% plot(x,y,'b+-');
figure;plot(xtest,ytest,'b-',xtest,actual_test,'r+');
xlabel('Xtest (m)');
ylabel('Ytest (m)');
h = legend('Desired output','Approximated curve',2);
set(h);
Xx=xtest;
Yy=actual_test;

```

6. Program KLF PID

```

function [KLFPID] = PIDKLF(ee,dee,iee)
e = (ee/500)-1;

```

```

de = (dee/500)-1;
ie = (iee/500)-1;

%deklarasi ef,def,ief
ef=zeros(3,1);
def=zeros(3,1);
ief=zeros(3,1);
%fuzzifikasi
%error karo error fuzzy
    if e<-1
        ef(1)=1;
    elseif e<0
        ef(1)=0-e;
        ef(2)=(e-(-1));
    elseif e<1
        ef(2)=1-e;
        ef(3)=e-0;
    else
        ef(3)=1;
    end
%error karo integral error fuzzy
    if ie<-1
        ief(1)=1;
    elseif ie<0
        ief(1)=0-ie;
        ief(2)=(ie-(-1));
    elseif ie<1
        ief(2)=1-ie;
        ief(3)=ie-0;
    else
        ief(3)=1;
    end
%error karo derivativ error fuzzy
    if de<-1
        def(1)=1;
    elseif de<0
        def(1)=0-de;
        def(2)=(de-(-1));
    elseif de<1

```



```

        def(2)=1-de;
        def(3)=de-0;
    else
        def(3)=1;
    end
%rule base boss mc vicar whelan
rb(1,,:)= [1 2 3; 2 3 3; 3 3 3];
rb(2,,:)= [1 1 2; 1 2 3; 2 3 3];
rb(3,,:)= [1 1 1; 1 1 2; 1 2 3];

%rule min
for i=1:3;
    for j=1:3;
        for k=1:3;
            v(i,j,k)=min([ef(i) ief(j) def(k)]);
        end
    end
end

%inference fuz
u=zeros(3,1);
for l=1:3;
    for m=1:3;
        for n=1:3;
            o=rb(l,m,n);
            u(o)=max(u(o),v(l,m,n));
        end
    end
end

%normalisasi
su1=0;
w1=0;
for a=1:3
    su1 = su1+u(a);
end
for b=1:3
    w1(b) = u(b)/su1;
end

```

```

%%deffuzifikasi
sigma_w1=0;
g2=0;
c1=[-1;0;1];
for c=1:3
    sigma_w1=sigma_w1+w1(c);
end
for d=1:3
    g1(d)=(w1(d)*c1(d));
end
for f=1:3
    g2=g2+g1(f);
end

output = g2/sigma_w1;
KLFPID = (output+1)*500;
end

```

RIWAYAT PENULIS



Michael Calvin Rendi Arianto adalah nama penulis dan akrab dipanggil Calvin. Penulis dilahirkan di Surabaya tanggal 14 Februari 1994, merupakan putra pertama dari dua bersaudara dari pasangan Soni Rumanto dan Rita Susanti. Penulis memulai pendidikan dari TK Santa Maria Blitar, SD Negeri Talun 3 Blitar, SMP Negeri 1 Wlingi Blitar, dan SMA Negeri 1 Talun Blitar. Setelah menyelesaikan pendidikan SMA pada tahun 2012, penulis melanjutkan pendidikan tinggi di Diploma Teknik Elektro ITS Surabaya tepatnya pada program studi Computer Control dan lulus pada tahun 2015. Selama kuliah di ITS, penulis banyak berkecimbung di dunia organisasi kemahasiswaan dan organisasi diluar kampus. Penulis Menjadi anggota Himpunan Mahasiswa D3 Teknik Elektro ITS dan menjadi asisten Laboratorium Elektronika Terapan di D3 Teknik Elektro ITS. Selanjutnya pada tahun 2015 penulis meneruskan pendidikan sarjana di Teknik Elektro ITS, tepatnya pada bidang studi Teknik Sistem Pengaturan. Pada bulan Juni 2017 penulis mengikuti seminar dan ujian Tugas Akhir sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Elektro dari Institut Teknologi Sepuluh Nopember Surabaya.

(halaman ini sengaja dikosongkan)